Toward Co-Robotic Construction:

Visual Site Monitoring & Hazard Detection to Ensure Worker Safety

by

Daeho Kim

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Civil Engineering)
in the University of Michigan
2021

Doctoral Committee:

Professor SangHyun Lee, Chair
Assistant Professor Jia Deng, Princeton University
Professor Vineet R. Kamat
Associate Professor Carol C. Menassa

Daeho Kim

daeho@umich.edu

ORCID iD: 0000-0002-7381-9805

DEDICATION

To my wife NaRi

ACKNOWLEDGEMENTS

Firstly, I would like to thank my advisor, Dr. SangHyun Lee, for all his help and guidance that he has given me during my MSE and PhD studies over the past five years, and I would like to thank my PhD committee members, Dr. Vineet R. Kamat, Dr. Carol C. Menassa, and Dr. Jia Deng for their thoughtful advice for my dissertation. I have been very fortunate to have been able to discuss my research with these great scholars.

Next, I would like to thank my colleagues. My colleagues who I spent countless hours with at the University of Michigan discussing our research in general: Dr. Meiyin Liu, Dr. Byungjoo Choi, Dr. Houtan Jebelli, Dr. Kwonsik Song, Dr. Dongmin Lee, Dr. Jinwoo Kim, Dr. Zuguang Liu, Gaang Lee, Hoyoung Lee, Francis Baek, Juhyeon Bae, Alejandro Newell, Ankit Goyal, Nurie Kim, and Julianne Shah. Their assistance, cooperation, and experience were essential for the completion of my PhD study.

Next, I also would like to thank my industry partners, WALSH Construction Co., TOEBE Construction LLC., and Barton Malow Construction Co., for helping me

with field data collection. There help and support were critical for the demonstration of my PhD study.

Last but not least, I would like to thank my family for their unwavering support and patience. This dissertation would not have been completed without their support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Construction has remained the least automated and productive as well as the most hazardous industry. Moreover, it has been plagued by a significant lack of diversity in its workforce as well as aging laborers. To address these issues, co-robotic construction has emerged as a new paradigm of construction. The industry is gradually gearing up to embrace robotic solutions, and many construction robots with various degrees of autonomy are under development or in the early stage of deployment. Presenting a different horizon of construction—harmonious co-existence and co-work between workers and robots—co-robotic construction is expected to reform labor-intensive construction into the more productive, safer, and more inclusive industry. However, an in-depth understanding of the robots' situational intelligence is still lacking, particularly conclusive logic and technologies to ensure workers' safety nearby autonomous (or semi-) robots, which is fundamental in realizing the co-robotic construction. To fill the gap, this research established a comprehensive robotic hazard detection roadmap and developed core technologies to realize it, leveraging unmanned aerial vehicles, computer vision, and deep learning. In this dissertation, I describe how the developed technologies with a conclusive logic can pro-actively detect the robotics hazards taking various forms and scenarios in an unstructured and dynamic construction environment. The successful implementation of the robotic hazard detection roadmap in co-robotic construction allows for timely interventions such as pro-active robot control and worker feedback, which contributes to reducing robotic accidents. Eventually, this will make human-robot co-

existence and collaboration safer, while also helping to build workers' trust in robot co-workers. Finally, the ensured safety and trust between robots and workers would contribute to promoting construction enterprises to embrace robotic solutions, boosting construction reformation toward innovative co-robotic construction.

CHAPTER 1

Introduction

1.1 Background

The construction industry has played a vital role in socio-economic development around the world. The industry has served societies with essential infrastructures for transportation, telecommunication, and energy and water supplies as well as with buildings, industrial plants, and housing, which are integral to sustaining an urbanized and industrialized society. Across the globe, the industry produces about $10 trillion worth of construction goods and services every year and employs more than 100 million people—equivalent to 13% of the world's gross domestic product and 8% of global employment, respectively (Mckinsey Global Institute 2017; International Labour Organization 2019).

Given the rapidly growing world population and booming urbanization, the construction industry's role and weight in sustaining urbanized societies will—needless to say—become more significant in the coming years. By 2050, the world population is projected to reach 9.7 billion, notably with the 70% of that population residing in urban areas (United Nation 2019). Needs for new infrastructures and for the rehabilitation of aging services in urban areas are, therefore, more salient than ever in addition to the need for new commercial and residential buildings. These projections lead to the conclusion that the construction industry should make a go of $97 trillion

worth of infrastructure projects worldwide by 2040 (Global Infrastructure Outlook 2020) while also providing 13,596 urban buildings (commercial=4,079; residential=9,517) every day through 2050 (Autodesk and Statista 2018).

Despite there being such a pressing demand for construction, the outlook for the construction industry's supply capacity is not that optimistic. The industry has been plagued by stagnant productivity—with minimal to no automation. There have been few innovations in construction workspaces and operations and the majority of construction work (e.g., from equipment operation to material handling) still heavily relies on workers' physical exertion as the primary source of production (Mckinsey Global Institute 2017). This heavy reliance on manpower has been taken for granted in the construction industry, leaving the industry behind in the race for productivity (Mckinsey Global Institute 2017). According to a survey conducted by the Mckinsey Global Institute, the labor-productivity growth of the construction industry averaged only 1% a year since 1995, compared with 2.8% for the total global economy and 3.6% for manufacturing (Mckinsey Global Institute 2017). Over the decade surveyed, more than 75% of construction firms globally failed to match the productivity growth compared to overall economies (Mckinsey Global Institute 2017). Absent change, the global need for infrastructures and urban buildings will be hard to meet according to the MGI survey (Mckinsey Global Institute 2017).

Granted its heavy reliance on manpower, poor workplace safety for field workers is another issue in the construction industry. According to a survey jointly conducted by the World Health Organization (WHO) and the International Labour Organization (ILO), the construction sector reported more than 60,000 fatal occupational injuries each year worldwide, contributing to about 17% of global occupational fatalities (WHO & ILO 2021). Such losses of life are a great sadness; additionally, they deal a heavy blow to construction projects. In the U.S. alone, the cost of fatal and non-fatal injuries in the construction industry is estimated to be $11.5 billion per year. The per-case cost is estimated to be $27,000, which is 80% higher than the average of other industries—in part due to how such cases delay project schedules, resulting in significant indirect losses (Waehrer et al. 2007).

On top of the above issues, the construction industry is also suffering from the shortage of skilled, young laborers. Prospective young workers consider construction trades to be low-compensation jobs. The total of young construction workers declined by around 30% from 2005

to 2016, with the turnover rate of 21.4%, unmatched by any other industries (Autodesk 2020). More than 90% of construction contractors are now having more than moderate levels of difficulties in finding skilled young workers (Commercial Construction Index 2019). With an aging workforce representing the majority of field workers, stagnant productivity and poor safety become even more critical issues for the global construction industry.

These impending problems faced by the construction industry are severely concerning. However, the construction industry is constantly seeking opportunities for innovation—now paying significant attention to co-robotic construction and its potential.

1.2 Emergent Co-Robotic Construction

The construction industry is gradually gearing up to embrace robotic solutions and reap the benefits of improved productivity and safety they can bring. Equipment manufacturers have retrofitted their equipment with autonomous kits and robotics companies are releasing a variety of construction robots that have varying degrees of autonomy. Swimming with this tide, construction academia is exploring new forms of robotic solutions. A variety of intelligent robots for a range of construction tasks are currently under development or are on their way to commercialization. To name a few, there are semi-automated bricklaying robots, 3D wall printing robots, rebar-tying robots, autonomous excavators, loaders, and trucks, and even humanoid construction robots (Figure 1.1). According to Verified Market Research, the global construction robot market that was valued at $212 million in 2018 is projected to reach $459 million by 2026, growing at a compound annual growth rate of over 10% (Verified Market Research 2018). Consequently, this projection leads to an estimate that at least more than 7,000 construction robots will be deployed in actual construction sites worldwide by year 2025 (Tractica 2019). Therefore, co-robotic construction is now more than a trend with field demonstrations for various construction robots now underway. While it is certain that construction robotics is still in its early stage compared to robotics for manufacturing, robots will be making their way into a variety of construction jobs in the not-so-distant future.

With the successful development and deployment of robotic solutions, future construction work will have a completely different horizon. This would be a space where human workers and

robots harmoniously co-exist. Most of physically demanding, highly repetitive, and unpleasant construction tasks would be taken over by robots while human workers focus on tasks that require fine dexterity, improvised decision making and troubleshooting to maneuver uncertainties, or supervising the robots. Tireless, precise, and consistent, these robots can carry out repetitive, laborious, and hazardous construction jobs strictly and quickly—an industry shift that is expected to improve construction productivity and safety by great degrees (Mckinsey Global Institute 2017; Devadass et al. 2019). Besides, as the major roles of construction workers shift from bodily-dominant tasks to more dexterous and intellectual ones, the construction workforce is expected to attract prospective workers from a range of demographics.

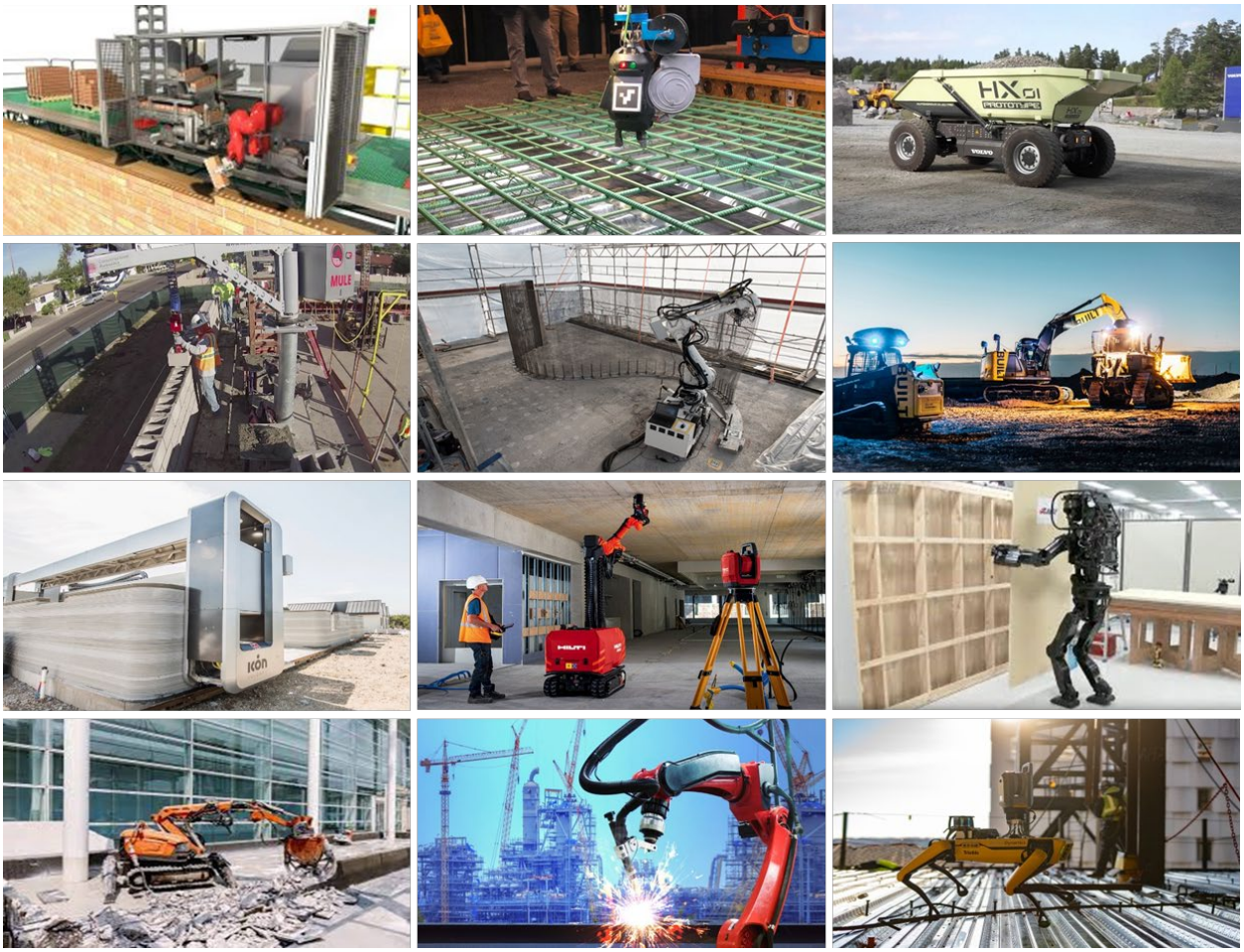

Figure 1.1 Examples of Upcoming Construction Robots

1.3 Problem Statement

The new horizon that co-robotic construction presents is promising. However, at de facto point of view, there is a pressing challenge in deploying intelligent robots with varying degrees of autonomy alongside field workers, particularly in unstructured and dynamic construction sites. My PhD research tackles this critical element in realizing co-robotic construction while also ensuring the safety of field workers. Workers' safety is an imperative factor that we must consider in realizing co-robotic construction. However, the dynamic nature of construction work and workspaces raises significant challenges to ensuring workers' safety.

Unlike a typical manufacturing line (e.g., the assembly line in an automotive factory), most construction takes place in outdoor environments that are highly dynamic and unstructured. The terrain of a construction site changes dramatically over time. The site layout itself is so unstructured that the route and movement of mobile resources (e.g., robots, equipment, and workers on foot) are erratic with boundaries frequently overlapping. In such environments, field workers often come in proximity to motorized resources and are thus exposed the risk of forcible collision in unforeseen ways (Kim et al. 2019a; Kim et al. 2019b; Kim et al. 2020). The number of construction worker fatalities caused by forcible collision is direct evidence of the magnitude of such hazards. According to the Census of Fatal Occupational Injuries conducted by the U.S. Bureau of Labor Statistics, 3,634 contact-driven fatalities were reported in the construction sector during 2009-2018, which accounted for about 41% of total construction fatalities (N=8,786) during that period (US BLS 2009-2018).

The main point here is that field workers will be assuming greater risk for such accidents occurring at co-robotic construction sites where they will be working alongside various types of mobile robots (Kim et al. 2020) (Figure 1.2). Any movement caused by a robot misperceiving a situation (e.g., approaching, deviating, and reversing) can pose a fatal threat to nearby workers. However, it is unknown how mobile robots' situational intelligence—such as their capacity for understanding, reasoning, and improvisational decision making—might rise to the dynamically evolving situations of construction sites (Figure 1.2). A mobile robots' navigation and behavior in such uncertain situations could include unexpected errors, which could pose a greater risk of forcible contact for nearby workers. Consider, for example, earthmoving robots such as an autonomous excavator, a loader, and a haulage truck moving around a construction site. Also

consider collaborative robots such as humanoid robots that work right next to human workers. These robots' routes and boundaries will overlap with field workers at times, raising the likelihood for accidents to occur (Figure 1.2).

Figure 1.2 Significance of Contact-Driven Accidents in Unstructured and Dynamic Construction Sites

Certainly, construction robots will be equipped with the safety functions like emergency stop and various ranging sensors such as Sound Navigation and Ranging (SONAR), Radio Detection and Ranging (RADAR), and Light Detection and Ranging (LIDAR). However, such sensors' measurement range, accuracy, and robustness cannot be guaranteed in unstructured outdoor construction sites where many sources of disturbance, such as frequent occlusions and unfavorable reflectors, are present (Ruff 2006; Kim et al. 2020). More importantly, these robots' intelligence and situational awareness will vary by agent and are bound to be mostly limited. Even though a robot's built-in safety functionalities might be significant, we cannot solely depend on them. Therefore, a conclusive and comprehensive hazard detection technology that can apply to any robot is needed.

1.4 Research Goal and Approaches

Against this background, the overarching goal of my PhD research is to develop and validate a visual site monitoring and hazard detection method that can complement the robots' built-in safety functionalities (Figure 1.3). To achieve this end, I leverage multiple imaging devices such as camera-mounted UAVs, camera-mounted hardhats, and pre-installed surveillance cameras. With these devices' multi-source streamed visual data, the method that I developed detects robotic hazards (i.e., contact-driven hazards) visually via deep neural network (DNN)-powered vision models. Not relying on each robot's built-in safety features, this method enables pro-active robotic hazard detection, providing another shield to prevent potential accidents that is easy and affordable. This method can provide any robot with augmented situational awareness regardless of their degree of autonomy and their level of intelligence. The digital cameras—for example, those mounted to a drone—will function as the robots' their third eye and the developed computational models can enhance their situational awareness. Proactive hazard detection as such allows for an immediate intervention, which would contribute to reducing the chance of robotic accidents.

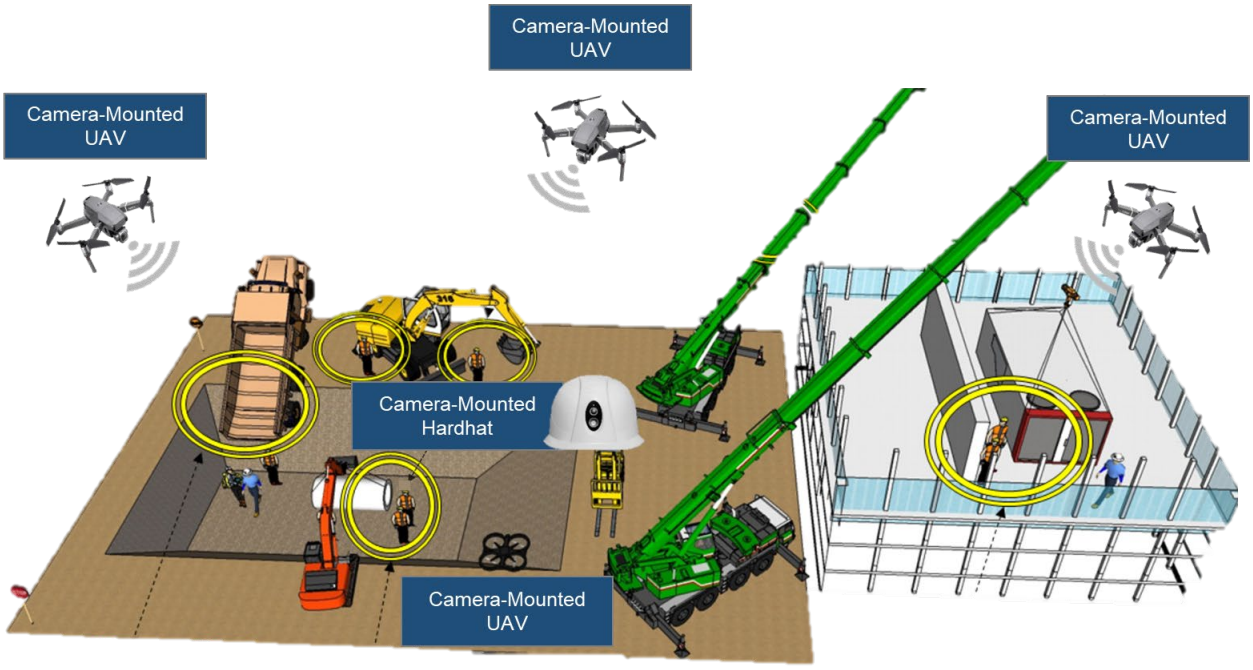

Figure 1.3 Overview of Visual Site Monitoring and Hazard Detection

In this research, to achieve a real-time, accurate, and scalable hazard detection method, I specifically adopt the approach of visual data analytics. Visual data analytics, also known as computer vision, is a field of creating artificial intelligence that can understand the visual world by interpreting a digital image (Patel and Patel 2020). The beauty of computer vision is in its diversification (Figure 1.4). Just from a single input source (e.g., an image or a video), we can draw diverse information such as where the target object is, how it poses and what action it is taking. Computer vision is multi-in-one: it deals with one input image (or a video) and figures all the surroundings out (Figure 1.4). Computer vision thus has great potential for understanding a total scene and, therefore, makes comprehensive hazard detection possible while also being more affordable than utilizing multiple physical sensors (Kim et al. 2020).

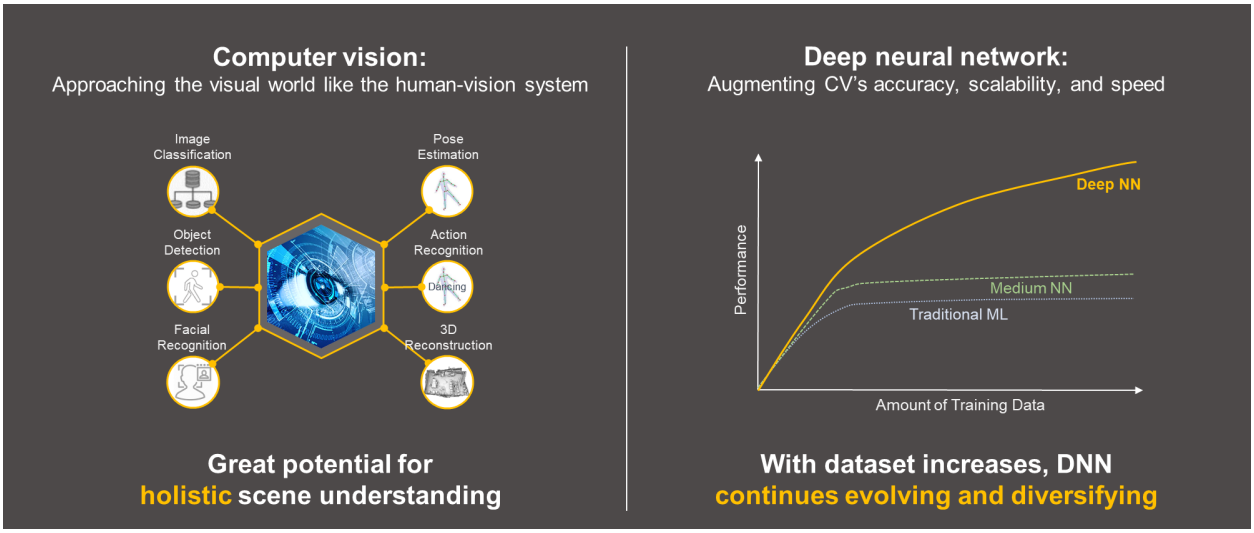

Figure 1.4 Potential of Computer Vision and Deep Learning as Multi-in-One Solution

Moreover, rapidly evolving deep learning greatly augments computer vision's capability. Deep learning can figure out the tricky feature extraction process of computer vision: with elaborate network architecture and supervised learning, it makes computer vision more accurate, faster, and more scalable. As is well-known, accessible public training datasets are increasing, DNN architectures and learning algorithms are being diversified, and hardware for parallel computing [e.g., Graphic Processing Unit (GPU)] is being enhanced. Such advancements are pushing the limits of DNN-powered computer vision (Figure 1.4). For these reasons, I saw the

potential of DNN-powered computer vision in developing a real-time, accurate, and scalable hazard detection method.

1.5 Robotic Hazard Detection Roadmap

There have been several research efforts using DNN-powered computer vision for construction sites and resource monitoring including hardhat detection (Wu et al. 2019), worker detection (Kim et al. 2016; Kim et al. 2018; Jeelani et al. 2021), equipment detection (Kim et al. 2017), worker pose estimation (Liu et al. 2017), and equipment pose estimation (Liang et al. 2019; Luo et al. 2020). However, there is a notable challenge in leveraging such DNN-powered vision models for the purpose of hazard detection.

The most significant challenge is in establishing a conclusive framework for the overall detection of robotic hazards. In order to achieve comprehensive robotic hazard detection, the monitoring techniques (e.g., object detection, pose estimation, etc.) must work as a whole by an inclusive logic because the hazards in construction take various forms (e.g., strikes by an autonomous vehicle or a robotic arm) and occur in scenarios where it is unfeasible to detect a hazard with a fragmentary scene understanding.

To address these challenges, I first developed a comprehensive robotic hazards detection roadmap as shown in Figure 1.5. This roadmap enables comprehensive detection of hazards by analyzing the following attributes between workers and robots in a phased manner: (i) proximity between workers and activated robots, (ii) (if they are in close proximity) semantic relation between workers and activated robots (i.e., whether they are co-working or not), and (iii) (if they are in close proximity and co-working) the co-worker's 3D pose. Each phase detects potential hazards that can be identified in three levels: safe, cautious, and hazardous. The roadmap has significance for enabling consistent collaboration between workers and robots while securing timely hazard detection.

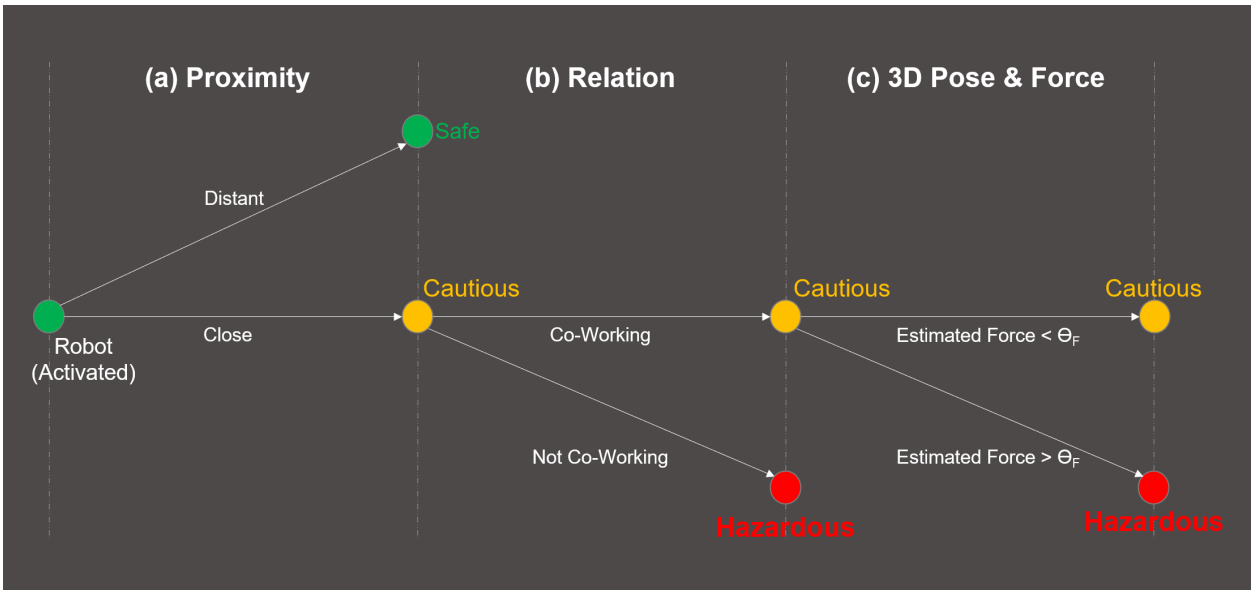

Figure 1.5 Hazard Detection Roadmap

Note: Θ_F stands for maximum allowable force of co-workers.

1. ***Proximity between Workers and Activated Robots (Figure 1.5(a)):*** In this roadmap, the first consideration is the proximity between a worker and an activated robot because there is a potential for collision. If a worker and a robot are situated at a safe distance apart, the condition is "safe." If the distance is not safe, then we must pay attention and stay "cautious" because there is potential for a collision.
2. ***Semantic Relation between Workers and Activated Robots in Close Proximity (Figure 1.5(b)):*** Even though a worker and an activated robot are in close proximity, there might not be a hazard present. Rather, we must take a look at the relation between the worker and the robot. If they are co-working (e.g., bricklaying where the worker finishes up the mortal joints while the robot piles up bricks), the proximity between them is safe and intentional. Likewise, proximity can be the precondition to a hazard but not the final determinant. Therefore, if the worker and robot are co-working, their relation is analyzed as "cautious" but not "hazardous" which would call for immediate control. If the worker and robot are in proximity and not working together, this roadmap classifies their relation as a "hazard."

3. ***Co-worker's 3D Pose (Figure 1.5(c)):*** Even though the worker and robot are co-working, there is still a chance of forcible collision, particularly by the robot's articulated body parts such as an arm. Therefore, it becomes important to monitor the worker's pose via 3D global coordinates in order to control the robot's movement and limit the potential contact force between the worker and robot so that it does not exceed the worker's maximum allowable force. If the robot's potential contact force is more than the worker's maximum allowable force, the roadmap classifies their relation as a "hazard" that needs immediate control. If not, though their relation is still "cautious," they can continue their work, and intervention would not be necessary. The roadmap classifies these two cases by analyzing the co-worker's 3D poses from the co-robot's viewpoint.

1.6 The Structure of the Dissertation

This dissertation is a compilation of the studies conducted to achieve core technologies that can operate the robotic hazard detection roadmap. This dissertation is composed of six chapters. Chapter 1 and Chapter 6 provide the introduction and conclusion of this work. Chapters 2 through Chapter 5 introduce each of the studies that correspond to the aforementioned topics. The following is the list of the chapters.

Chapter 1: Introduction. This chapter introduces the background, problem statements, and objectives and approaches of the entire research effort.

Chapter 2: Real-Time Proximity Monitoring between Workers on Foot and Activated Mobile Robots using Camera-Mounted UAVs. This chapter presents a real-time proximity monitoring method using camera-mounted UAVs, a DNN-powered object detection model, and a distance measurement method via image rectification. The validations of the object detection and distance measurement modules are presented. Lastly, the field test result is presented.

Chapter 3: Proximity Prediction using a Conditional Generative Adversarial Network. This chapter presents the follow-up study to proximity monitoring, which is proximity prediction.

The study introduces a conditional generative adversarial network (GAN)-based proximity prediction method, which uses detected objects' past trajectories as input. The validation result of the conditional GAN-based trajectory prediction module and field test results are presented.

Chapter 4: Semantic Relation Detection between Workers and Robots using a One-Stage Two-in-One DNN. This chapter presents a one-stage DNN-powered vision model that can infer semantic relations between workers and robots. Validation of the developed model and test results are presented.

Chapter 5: 3D Pose Estimation of Co-Workers using a Synthetic Construction Data-Trained 2D-to-3D Pose Transfer DNN. This chapter presents a DNN model that can infer a worker's 3D pose from a given 2D pose. In particular, this chapter introduces a novel way to use synthetic construction data for training the DNN. A detailed process for developing the model with synthetic data and the best model's performance are presented.

Chapter 6: Conclusions. This chapter provides a summary of the conclusions that can be drawn from the research. Future research agendas for co-robotic construction are also provided.

CHAPTER 2

Real-Time Proximity Monitoring Between Workers on Foot and Active Mobile Robots using Camera-Mounted UAV[1]

2.1 Introduction

This study is aimed to address the first agenda of the robotic hazard detection roadmap: proximity monitoring between workers and activated (mobile) robots. Proximity is an absolute precondition for a potential collision, which must be monitored first to determine whether a situation is categorized as "safe" or "cautious" (i.e., whether a worker is in the action radius of a robot or not). To achieve less-occluded, real-time, and accurate proximity monitoring in this study, I specifically leveraged camera-mounted UAVs as imaging devices and developed a real-time visual proximity monitoring method leveraging DNN-powered object detection and an image rectification technique that allows for the measurement of real distances on ground.

Early research on proximity monitoring has been dominated by a wide range of tag-based sensors including Radio Frequency Identification (RFID) (Teizer et al. 2010; Marks and Teizer

[1] This chapter is adopted from Kim, D., Liu, M., Lee, S., and Kamat, V.R. (2019) "Remote Proximity Monitoring between Mobile Construction Resources using Camera-Mounted UAVs." *Automation in Construction*, 99(2019), 168-182 and Kim, D., Lee, S., and Kamat, V.R. (2020) "Proximity Prediction of Mobile Objects to Prevent Contact-Driven Accidents in Co-Robotic Construction." *Journal of Computing in Civil Engineering*, 34(4), 04020022.

2012), Magnetic Field (MF) (Teizer 2015), Global Positioning System (GPS) (Ruff 2001), and Bluetooth Low Energy (BLE) (Park et al. 2016; Park et al. 2017). However, implementation of this sensor-based application may be challenged in practice. For example, the prerequisite that all entities should have attached sensors could be burdensome for both contractors and workers. Such a prerequisite could be costly and hard to manage in projects where tremendous volumes of personnel, robots, equipment, and materials are involved (Park and Brilakis 2012; Memarzadeh et al. 2013; Kim et al. 2016; Kim et al. 2017) and could be further intrusive to workers who do not want to be tagged (Brilakis et al. 2011; Park and Brilakis 2012). Besides, sensing ranges could be affected by various factors such as ambient conditions or approach angles and speeds since the sensors operate based on wave signal propagation (Teizer 2015; Park et al. 2016; Park et al. 2017).

Certainly, many construction robots, those under development or being prototyped, are equipped with Time-of-Flight (ToF)-based sensors such as Sound Navigation and Ranging (SONAR), Radio Detection and Ranging (RADAR), and Light Detection and Ranging (LIDAR). However, such sensors' measurement ranges, lines of sight, accuracy, and robustness cannot be assured in outdoor construction sites where there are many sources of disturbance such as frequent occlusions (Kim et al. 2020). In construction sites, it is common and frequent that materials are stacked around or moved from place to place, temporary structures are installed, and motorized resources move around. In such a dynamic environment, the ToF-based sensor's line-of-sight can often be occluded by moving objects or temporary structures, limiting the robot's field of vision.

To address such issues, I leveraged camera-mounted UAVs as imaging devices and solved proximity monitoring by a vision method. An ordinary camera-mounted UAV can capture moving entities continuously while accessing hard-to-reach areas (Ham et al. 2016). This mobility allows for the monitoring of wide areas, specifically those that are less occluded, which is not feasible to achieve with conventional imaging devices such as surveillance or portable cameras (Zollmann et al. 2014; Michael et al. 2014; Lin et al. 2015; Han et al. 2015). The digital cameras, mounted to UAVs, can be a robots' third eye and my developed vision method can detect multiple objects and measure distances between them simultaneously in real-time. This type of visual monitoring has the potential to enable cost-effective and non-invasive proximity monitoring while complementing existing sensing technologies.

To this end, my research specifically tackled two technical challenges facing computer vision techniques, object localization and distance measurement, which are fundamental to visual proximity monitoring. Real-site videos entail uncertain variations. For example, each frame involves different viewpoints, scene scales, and illumination. Also, each entity (i.e., workers, robots, or equipment) has an individually distinctive appearance. These variations impose restrictions on the localization capability of hand-crafted algorithms since they operate as designed and thus could not be adaptive (Brilakis et al. 2011; Park and Brilakis 2012; Memarzadeh et al. 2013; Kim et al. 2016). In addition, measuring distance on a 2D image is extremely challenging due to the lack of depth information (i.e., the 3rd coordinate of a point). Therefore, accurate distance measuring requires post-processing for 3D reconstruction which, in turn, requires a significant amount of computational cost.

To overcome these challenges, my research was conducted in two thrusts: (i) the application of a deep neural network [i.e., YOLO-V3 (Redmon and Farhadi 2018)] for robust object localization and (ii) the development of an image rectification method that enables measurement of actual distances on a 2D image without 3D reconstruction. This research tested the developed method on real-site aerial videos so as to evaluate its monitoring performance in real scene settings. This chapter provides details of the developed method and test results as well as the results' implications and future research directions.

2.2 Existing Sensor-based Technologies for Proximity Monitoring

Based on operation principles, proximity sensors can be largely categorized into two types: (i) Time-of-Flight (TOF)-based sensor and (ii) tag-based sensor. The TOF-based sensor, installed on a robot, measures the distance of surroundings (e.g., geographic features, obstacles, and workers) by emitting a certain form of energy and reading its time-of-flight. As well-recognized sensors, SONAR, RADAR, and LIDAR are included in this category.

SONAR (or ultrasonic sensor) measures distances to physical objects by transmitting a high-frequency sound wave and measuring the TOF of its echo reflected from the target objects. A sound wave requires a certain medium to travel. Its propagation, therefore, involves many disturbances by the medium's physical conditions (e.g., temperature and pressure), and it can be

more so particularly in the case of longer-range detection (Varghese and Boone 2015). Accordingly, the application of SONAR in mobile robots has been limited to short-range detection—typically less than 3 meters (e.g., reverse parking) (Ducarme 2019).

On the other hand, RADAR uses radio signal (300 MHz - 40 GHz), a kind of electromagnetic wave, which does not require a certain medium to travel. It thus functions in many wild conditions (e.g., rain, fog, snow, and dust) and has a long-range of reading—generally more than 30 meters (Ducarme 2019). In addition, using Doppler Effect (Chen et al. 2006), it can also detect the speed of moving objects as well as its proximity (Varghese and Boone 2015). However, the performance of RADAR can vary by reflectors. This is because the radio signal can be easily dispersed, particularly when encountering unfavorable reflectors such as plastics, dry wood, or objects with large flat surfaces (Ruff 2006).

LIDAR also uses a kind of electromagnetic wave, the beam of light (or laser). It is able to not only measure distances to objects but also scan 3D surroundings with multi-axis lasers. The more lasers a LIDAR transmits, the denser 3D world can be reconstructed (Ducarme 2019; Varghese and Boone 2015). Of stand-alone sensors, LIDAR is often cited as the most accurate proximity sensor (Gargoum et al. 2018). Also, the 3D readout is potentially used as the primary source for the path planning of many autonomous navigating robots (Kim et al. 2018). However, LIDAR, as with other TOF-based sensors, cannot distinguish what the detected objects are. To distinguish objects, it needs additional object classification software (Ducarme 2019).

Distinctive to these TOF-based sensors, tag-based sensors utilize an energy field (e.g., electromagnetic field) and detect proximity via the signal communication between a reader mounted to a robot and tags worn by workers. With this principle, many kinds of sensors have been devised, including radio frequency identification (RFID), magnetic field (MF), and Bluetooth low energy (BLE). As the tag-based sensors don't rely on the TOF measurement, they are less affected by the line-of-sight (Ducarme 2019). However, the tag-based sensors have hardly gained a competitive edge over the TOF-based sensors in terms of accuracy and fidelity. According to a test conducted by Park et al. (2016), the proximity errors of RFID, MF, and BLE sensors were up to 5.0, 3.4, and 2.6 meters, respectively, with the standard deviation of 2.1, 0.3, and 1.8 meters. Although the tag-based sensors still have the potential to complement other technologies (e.g.,

SONAR, RADAR, and LIDAR), the prerequisite that all targets need to be attached with a tag hinders their application in construction (Memarzadeh et al. 2013; Park et al. 2012).

The proximity sensors have been widely applied in robotics to assist the robots' collision avoidance (Cui et al. 2019). However, the effectiveness, availability, and functionality of the existing proximity sensors could be challenged in a highly unstructured and dynamic construction site. For example, the TOF-based sensors (e.g., SONAR, RADAR, and LIDAR) could be frequently blinded by physical barriers; while the performances of tag-based sensors (e.g., RFID, MF, and BLE) are susceptible to deterioration due to the jamming caused by metallic or wooden objects, both of which are common in construction sites.

2.3 Existing Vision-based Technologies for Proximity Monitoring

Over recent years, computer vision-based methods have demonstrated great potential as a supplementary technology to proximity sensors (Zhu et al. 2017; Park et al. 2016; Memarzadeh et al. 2013; Park et al. 2012; Brilakis et al. 2011). It uses one or more imaging devices (e.g., digital camera) to capture multiple targets and stream the digital images to a computer. In turn, it utilizes the computing power to conduct object detection and proximity measurement. With the improvement of computing power, the potential of the computer vision continues to grow. This growth is evidenced by the number of construction studies that have explored computer vision-based proximity monitoring technologies. For example, Memarzadeh et al. (2013) developed an algorithm to detect multi-class construction objects by integrating histogram of oriented gradient (HOG) and histogram of hue-saturation-value (HSV); Kim et al. (2016) proposed a proximity monitoring framework that employs Gaussian mixture model (GMM)-based object detection; Kim et al. (2017) introduced another proximity monitoring framework using multi-view cameras and object detection based on HOG and support vector machine (SVM). The previous studies have greatly contributed to examining the potential of computer vision-based proximity monitoring technologies. However, there are several drawbacks of the computer vision-based methods, which need to be addressed for construction applications.

2.3.1 Limited field of view of stationary imaging devices

A major imaging device widely used is stationary cameras such as tripod-mounted or surveillance cameras (Zhu et al. 2017; Park et al. 2016; Brilakis et al. 2011). These cameras are cheap, readily available, and easy to apply. However, this technology can involve frequent occlusions of targets (i.e., the situation that targets are occluded by physical barriers and so become invisible) particularly on construction sites where a number of obstacles to the camera's line-of-sight are scattered (Kim et al. 2019b). The problem is that such occlusions are fatal to any computer vision-based object detection because the computer vision is bound to rely on the visible information of a target (e.g., the target's pixel values and configuration). Therefore, the application of mobile imaging devices which have a wider line-of-sight and mobility, thereby reducing such occlusions (e.g., UAVs), must be considered.

2.3.2 Low speed and accuracy of object detection

Many earlier studies applied one or more hand-crafted features—such as Histogram of Gradient (HOG), Scale Invariant Feature Transform (SIFT), and Speeded-Up Robust Features (SURF)—to object detection. However, using such features naturally involves a heavy computation due to pre-processing and multiple steps for feature extraction, resulting in slow processing speed (Kim et al. 2019b). In addition, although the hand-crafted features could work well in a customized imaging condition (e.g., controlled viewpoint, scale, and illumination), they could lose their representative power for a target in unconformable conditions—such as viewpoint variation, scale variation, illumination variation, background clutter, or intra-class variation (Brilakis et al. 2011; Park and Brilakis 2012; Memarzadeh et al. 2013; Kim et al. 2016). For example, the object detector that uses HOG could fail to detect same object if a huge illumination difference occurs, while the one that uses SIFT could fail to detect equipment with a distinctive appearance. Therefore, the higher-level of representation of a target is required in localizing construction entities on a UAV-captured video where the dynamic viewpoint gets to amplify such variations.

Recently, DNN-based object detection has made large progress in terms of speed and accuracy by leveraging parallel computing and finer-level learned features. Accordingly, an

increasing number of studies have attempted to apply the DNN-based object detection framework for construction applications. For example, Fang et al. (2018), Luo et al. (2018), Son et al. (2019), and Yan et al. (2019) applied Faster Region-based Convolutional Neural Network (Faster R-CNN, Ren et al. 2017) for construction objects detection; Kim et al. (2018) and Alipour et al. (2019) applied Region-based Fully Convolutional Network (R-FCN, Dai et al. 2016). The studies applying DNNs proved to greatly improve the speed and accuracy of construction object detection. However, since the DNNs (i.e., Faster R-CNN and R-FCN) rely on two-stage inferences (region proposal and classification) by two separated networks, they involve a high computational cost and couldn't achieve the real-time operation—30 frame per second (FPS). The real-time operation is definitely critical in assisting timely proximity monitoring. Computer vision-based methods must demonstrate real-time operation for real-world applications.

2.3.3 Lack of distance measurement techniques on a 2D image

Proximity monitoring is completed by measuring the straight-line distances among targets, which can be straightforward given 3D spatial information. However, using a 3D sensing device (e.g., stereo-vision camera and RGB-D sensor) may not be much viable for onsite operation due to its limited sensing range and the vulnerability to outdoor conditions (Chi and Caldas 2012; Seo et al. 2015). For example, stereo-vision camera (e.g., Bumblebee XB3, Point Grey Research, Inc.) is restricted to low resolutions and requires a significant amount of computational cost (Seo et al. 2015). Also, RGB-D sensor (e.g., MS KinectTM) and Flash LADAR are susceptible to sunlight as well as have restricted measuring range (i.e., 5 meters and 10 meters, respectively) (Chi and Caldas 2012; Seo et al. 2015).

In construction, there have been few studies attempting to monitor the proximity among construction entities on a 2D image (Kim et al. 2016; Kim et al. 2017). These studies estimated proximity by measuring pixel distances among detected objects and used the value in evaluating workers' safety level. Although the pixel distance can be useful in determining relative safety level, it would not be able to represent the real scale of distance due to the lack of depth (i.e., scene scale) and the projective distortion. To be more specific, an ordinary camera maps 3D real space onto a 2D image plane through its monocular lens by perspective projection. During this compressive

process, the depth information is lost, and projective distortion occurs, making the original properties of a scene (e.g., length, area, length ratio and area ratio, angle, and parallelism) and proximity distorted.

On this problem, several studies presented post-processing as a solution to recover the depth information (i.e., the lost 3rd coordinate). Brilakis et al. (2011) proposed a triangulation framework using multiple 2D cameras to determine the 3D coordinates of construction resources whereas Yang et al. (2013) attempted another triangulation algorithm, i.e., Structure from Motion (SFM), for 3D reconstruction. Although the recovered depth information enables distance measurement, such epipolar geometry-based post-processing requires a significant amount of computational cost for extracting features, calculating fundamental matrix, and lastly triangulation (Seo et al. 2015). Moreover, this triangulation is viable only if the camera's extrinsic parameters (i.e., location and orientation) and feature matching are given at a very precise level. Hence, this 3D reconstruction technique may not be the best choice for onsite proximity monitoring, specifically in the context of a mobile UAV.

2.4 Research Objectives

The methods to date have shown a potential of visual localization and distance measurement on an image, but have not yet reached the capability to be used for onsite proximity monitoring. The localization techniques may not be sufficiently robust against casual variations of real scenes. In addition, the 3D reconstruction would not be a viable option for proximity monitoring on account of its massive computations and sensitivity to given parameters (e.g., camera's location and orientation).

With these challenges, the objective of this research is to achieve (i) automated, fast, and robust localization of construction entities, and (ii) cost-effective but reliable distance measurement directly from a 2D image. Toward these ends, this research conducts two research thrusts: (i) the application of a deep neural network, i.e., YOLO-V3 (Redmon and Farhadi 2018) to object localization; and (ii) the development of an image rectification method that allows of measuring actual distance on a 2D image without the 3D inference (Figure 2.1). In the following

two sections, the details on the proposed methods are explained with the test result. In succession, tests on aerial construction site videos and discussions on the test result will follow.

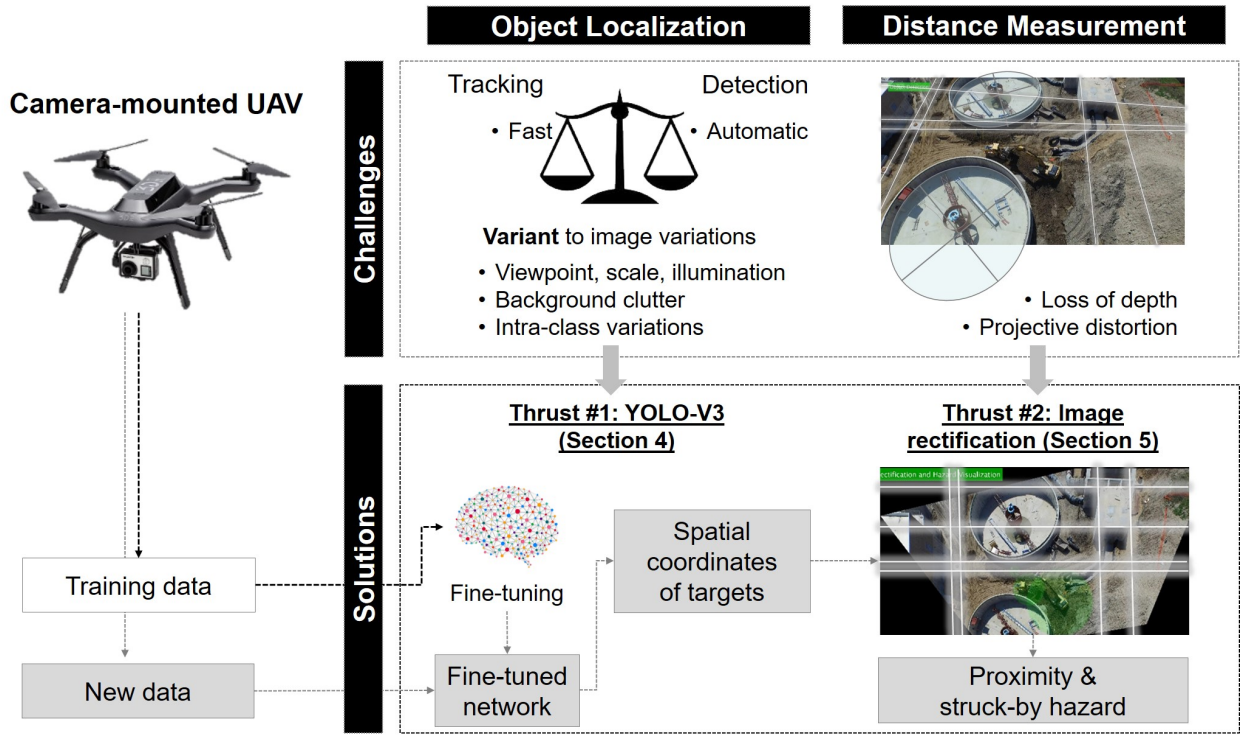

Figure 2.1 Two Research Thrusts for UAV-Assisted Visual Proximity Monitoring

2.5 Thrust #1: YOLO-V3 for Object Localization

Recently, DNNs have demonstrated superior performance in object detection, overcoming the detection challenges across the computer vision community—such as COCO detection challenges (Table 2.1). The deep networks enable the extraction of fine-grained features, which have demonstrated a more robust operation in the object detection (Girshick et al. 2015; He et al. 2015; Girshick 2015; Ren et al. 2017; Fang et al. 2018; Kim et al. 2018; Kolar et al. 2018). At the same time, the DNNs have substantially reduced their computational costs as well with the advancement of computing mechanism (e.g., parallel computing) and hardware [e.g., graphical processing unit (GPU)] (Fang et al. 2018; Redmon and Farhadi 2018). Table 2.1 shows state of the art DNNs for object detection and their performances [i.e., mean average precision (mAP) and frame per second (FPS)] on the COCO benchmark dataset (Redmon and Farhadi 2018).

In construction, there have been several efforts to use the DNNs for the localization of construction entities. For example, Fang et al. (2018) attempted to detect non-hardhat-use using Faster R-CNN; Kim et al. (2018) applied R-FCN for detecting equipment in tunnel construction; on the other hand, Kolar et al. (2018) designed a customized DNN by combining a VGG-16 (i.e., feature extractor used in Faster R-CNN) and a Multi-Layers Perception (MLP) network for safety guardrail detection. Evidently, these studies showed the successful introduction of the DNNs to construction research, validating its detection performances (e.g., mAP) on construction data. For this study, however, the Region Proposal Network (RPN)-based DNNs—such as Faster R-CNN or R-FCN—would not be the best option due to their high computational cost (Figure 2.2). As shown in Table 2.1 and Figure 2.2, the FPS for the Faster R-CNN (i.e., 17 FPS) and R-FCN (i.e., 12 FPS) are insufficient for real-time operation (i.e., 30 FPS).

Table 2.1 State of the Art DNNs for Object Detection: Performance on COCO Dataset

Model	Train dataset	Test dataset	mAP	FPS
Faster R-CNN	COCO train-val	COCO test-dev	42.70%	17
SSD321	COCO train-val	COCO test-dev	45.40%	16
DSSD321	COCO train-val	COCO test-dev	46.10%	12
R-FCN	COCO train-val	COCO test-dev	51.90%	12
Retinanet-50-500	COCO train-val	COCO test-dev	50.90%	14
YOLO-V2	COCO train-val	COCO test-dev	48.10%	40
YOLO-V3	**COCO train-val**	**COCO test-dev**	**55.30%**	**35**

In this sense, this study applies YOLO-V3 (Redmon and Farhadi 2018) that allows for real-time operation (i.e., 35 FPS) as well as state of the art detection performance (i.e., 55.3 mAP on COCO dataset, Table 2.1). The YOLO-V3 doesn't require an additional step for region proposal (Figure 2.2). Instead, it realizes the convolutional implementation of sliding window during its operation, thereby making one-stage detection and real-time operation possible (Redmon and Farhadi 2018). With this advantage, YOLO-V3 could afford to have a deeper convolutional network and thus achieve the state of the art performance on object detection (Figure 2.2).

The published YOLO-V3 network, pre-trained with ImageNet, are not learned from the construction contexts such as construction equipment, workers, and backgrounds. Furthermore,

this network will not be compatible with UAV-captured images because they are not experienced with aerial viewpoints. For example, a human in a UAV-captured image has a completely different appearance and scale than one in ImageNet, which must puzzle the convolutional layers and deteriorate the localization performance in the end. On the other hand, to train the network with construction data from scratch must involve a significant risk of overfitting due to the imbalance between the network capacity and the amount of training data. Therefore, this research elects transfer learning to avoid the potential of overfitting as well as to fine-tune the published network to construction settings successfully.

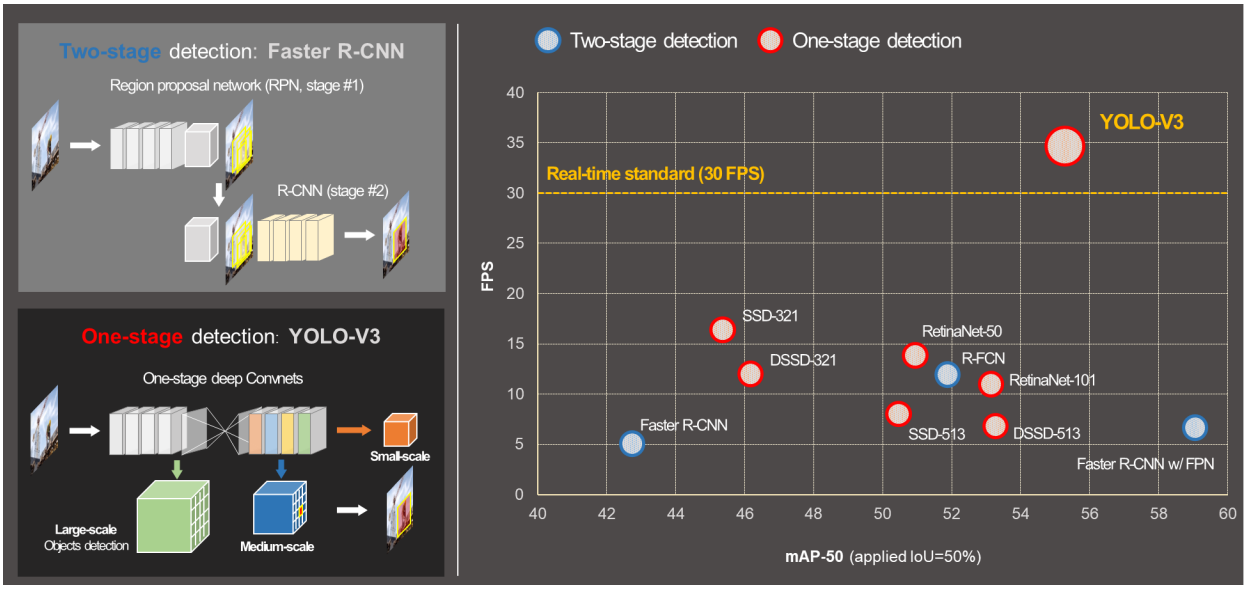

Figure 2.2 Object Detection DNNs' Accuracy and Speed: One-Stage vs. Two-Stage

2.5.1 Network description

The YOLO-V3 consists of two main networks: (i) feature extractor and (ii) object detector (Figure 2.3).

- Feature extractor (from 1st to 75th layer): the first network, called darknet-53, takes a resized image (416x416x3) as an input and outputs a 3D feature tensor (13x13x1024).

The darknet-53 has a deep architecture with successive 52 convolutional layers (i.e., 1x1 or 3x3), which can extract fine-grained features from a coarse data. In particular, this network incorporates residual skip connections in the intervals of two convolutional layers (i.e., total 23 shortcut layers). The connection initially devised for a residual network helps the darknet-53 to deals with the vanishing gradient problem occurring while training by residually propagating previous features into forward.

- Object detector (from 76th to 107th layer): the second network takes the 3D feature tensor (13x13x1024) and makes detections. The uniqueness of this network resides in its ability to achieve detection at three different scales, thereby improving scale invariance. This network gradually widens the feature tensor from 13x13 to 26x26, and 52x52 through upsampling and concatenation layers. Meanwhile, three branches come out and each makes a final feature tensor at the different scale (i.e., 13x13x42, 26x26x42 and 52x52x42 at 82nd, 94th, and 106th layer, respectively). Each final feature tensor is then fed into YOLO layer that classifies object label with class-wise logistic regressions and localizes objects with bounding box regressors.

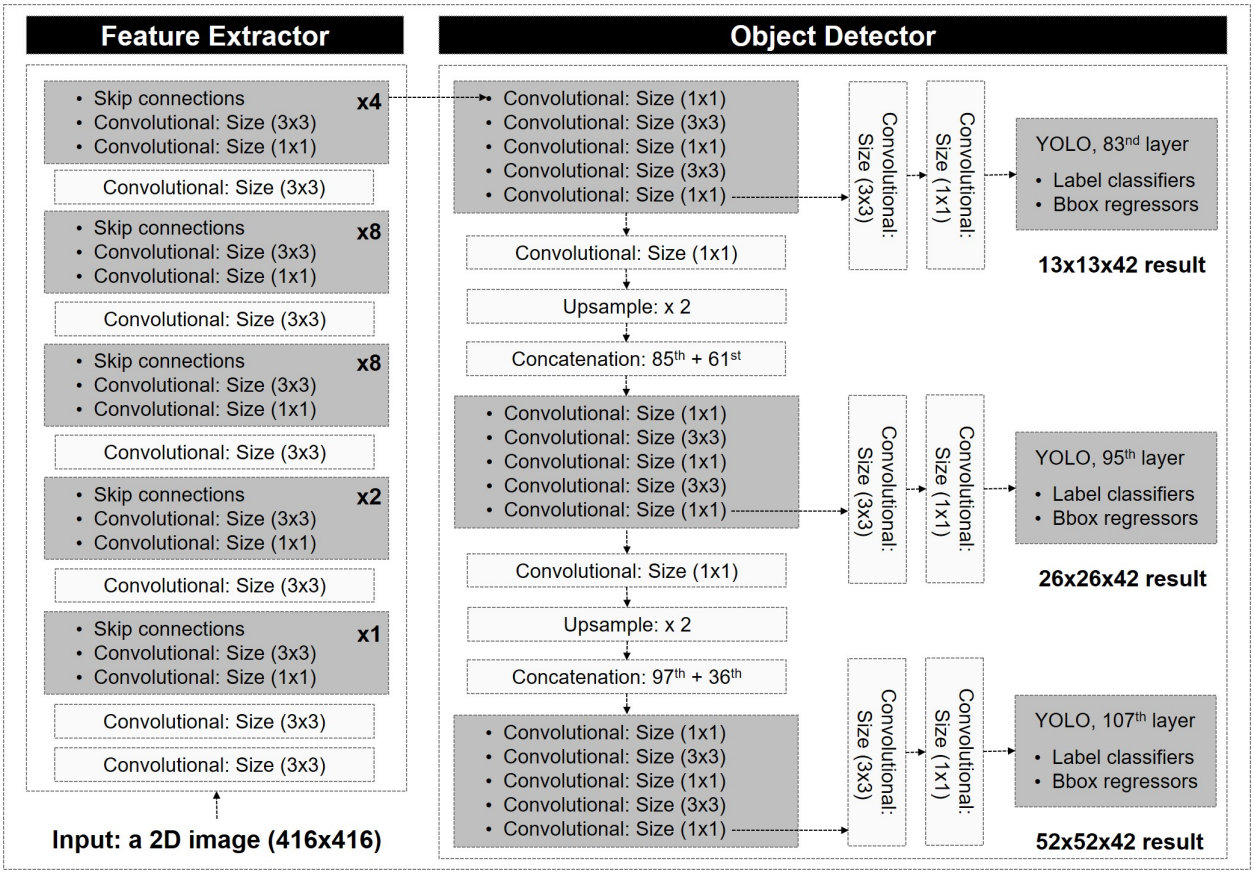

Figure 2.3 Architecture of YOLO-V3

2.5.2 Test result

The total of 4,512 frames capturing construction workers and equipment were extracted from construction site videos and labeled as shown in Figure 2.4. Of these, 4,114 images were used for the fine-tuning and the other data, 398 consecutive images (i.e., a section of a UAV video), were used for testing. This test considered the three types of object classes: (i) construction worker; (ii) wheel loader; and (iii) excavator (Figure 2.4).

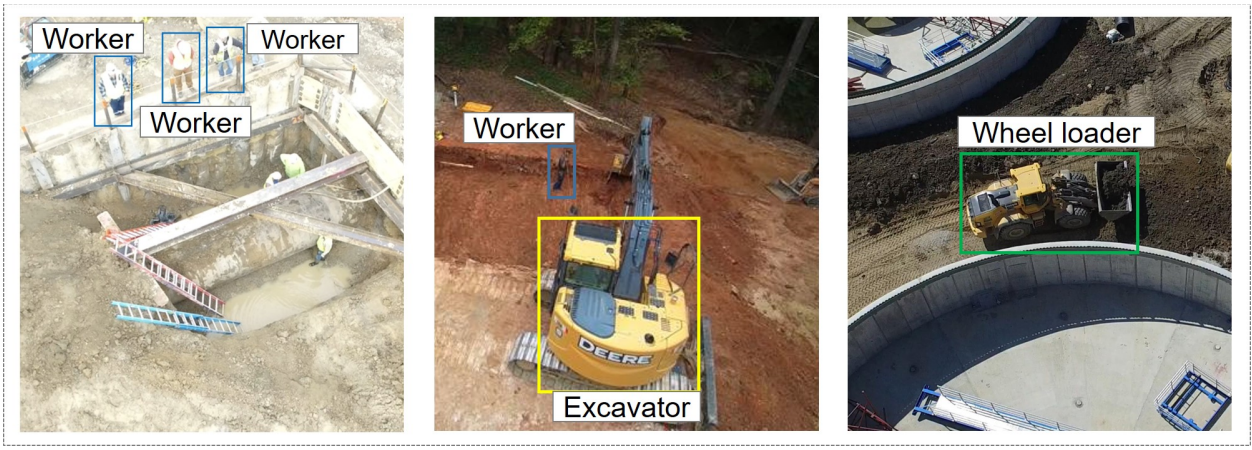

Figure 2.4 Examples of Training Dataset and Labels

The first role of the YOLO-V3 in proximity monitoring is to make correct object detections. To test the detection performance of the fine-tuned network, this test uses mean average precision (mAP, Equation 2.1) and average intersection over union (IoU, Equation 2.2), which are the typical evaluation metrics used for detection challenges—such as PASCAL VOC and COCO. As shown in Table 2.2, the tuned network could reach to acceptable mAP and average IoU: (i) mAP=90.82% and (ii) average IoU=80.97% in this test.

$$\text{mAP} = \frac{1}{n} * \sum_{1}^{n} \left(\frac{1}{11} * \sum_{r=0.0,0.1,\ldots,1.0} AP_r \right) \qquad \text{Equation 2.1}$$

Note: n stands for the total number of object classes; AP_r stands for maximum precision at a certain recall value r (i.e., 0, 0.1, 0.2, …, 1.0).

$$\text{Average IoU} = \frac{1}{k} * \sum_{1}^{k} \left(\frac{AoO}{AoU} \right) \qquad \text{Equation 2.2}$$

Note: k stands for the total number of detected objects; AoO stands for area of overlap; AoU stands for area of union.

Table 2.2 Result of Object Detection by YOLO-V3: mAP and Average IoU

# Iter.	Average precision			mAP	Average IoU	Average precision
	Excavator	Worker	Wheel loader			Reference object
500	14.01%	0.00%	17.79%	10.60%	0.00%	0.17%
600	27.04%	11.86%	42.62%	27.17%	38.83%	67.98%
700	56.97%	63.86%	62.87%	61.23%	38.77%	57.37%
.	.	.	.	.	.	.
.	.	.	.	.	.	.
1000	83.48%	80.48%	85.77%	83.24%	60.99%	89.43%
1100	90.71%	90.57%	82.65%	87.98%	69.00%	90.36%
1200	89.05%	86.98%	79.05%	85.03%	63.72%	87.04%
.	.	.	.	.	.	.
.	.	.	.	.	.	.
10000	90.84%	90.63%	90.79%	90.75%	77.16%	90.91%
10100	90.77%	90.73%	90.82%	90.77%	78.18%	90.86%
10200	90.79%	90.73%	90.79%	90.77%	78.65%	90.84%
.	.	.	.	.	.	.
	.	.	.	.	.	.
19800	**90.77%**	**90.86%**	**90.84%**	**90.82%**	**80.97%**	**90.86%**
19900	90.77%	90.76%	90.84%	90.79%	80.36%	90.86%
20000	90.75%	90.84%	90.82%	90.80%	78.82%	90.84%

Note: mAP and average IoU are for excavator, worker, and wheel loader; reference object is the material to be used for image rectification whose role and function will be detailed in the next section.

In proximity monitoring, it is also critical to find the correct location for detected objects (i.e., object-centered coordinates). Hence, this test further evaluates the fine-tuned network by the Average Localization Error (ALE) (i.e., the average of the Euclidean distance between ground truth position and estimated position, Equation 2.3). As shown in Figure 2.5, the fine-tuned network showed promising localization performance, tracking ground truth consistently with the acceptable ALE: (i) worker=0.16 meters; (ii) wheel loader=0.37 meters; and (iii) excavator=0.31 meters (Table 2.3).

$$\text{ALE} = \frac{1}{n} * \sum_{i=1}^{n} SF\sqrt{(x_{gt} - x_e)^2 + (y_{gt} - y_e)^2} \qquad \text{Equation 2.3}$$

Note: n stands for the total number of frame; SF stands for the scale coefficient that converts pixel distance to the metric unit (i.e., meter); x_{gt} and y_{gt} stand for coordinates of ground truth; and x_e and y_e stands for the estimated coordinates.

Table 2.3 Result of Object Localization by YOLO-V3: ALE

Frame #	Estimated coordinates						Localization error (unit: meters)		
	Worker		Wheel loader		Excavator		Worker	wheel Loader	Excavator
	X	Y	X	Y	X	Y			
1	203	114	262	156	362	191	0.14	0.32	0.37
2	203	115	260	156	361	191	0.16	0.15	0.27
3	203	115	261	156	361	191	0.21	0.24	0.33
4	204	115	260	156	361	191	0.17	0.17	0.17
5	204	115	260	156	361	191	0.12	0.27	0.18
6	203	115	260	156	361	192	0.14	0.35	0.06
7	204	115	259	156	360	192	0.04	0.22	0.14
8	203	115	259	156	360	192	0.09	0.21	0.18
9	203	116	259	156	359	192	0.03	0.18	0.13
10	203	116	258	156	359	192	0.08	0.17	0.17
						.			
						.			
						.			
						.			
389	211	130	92	126	228	234	0.19	0.45	0.26
390	211	130	92	126	227	234	0.15	0.43	0.24
391	211	131	91	126	228	234	0.16	0.44	0.25
392	212	130	90	126	227	233	0.19	0.34	0.27
393	212	130	90	124	226	234	0.11	0.29	0.13
394	211	130	90	124	226	233	0.04	0.25	0.21
395	212	130	90	123	226	233	0.09	0.26	0.09
396	212	130	89	123	226	233	0.08	0.23	0.02
397	212	129	89	123	226	233	0.18	0.21	0.08
398	213	129	88	122	225	233	0.14	0.11	0.13
Average localization error (ALE, unit: meters)							**0.16**	**0.37**	**0.31**

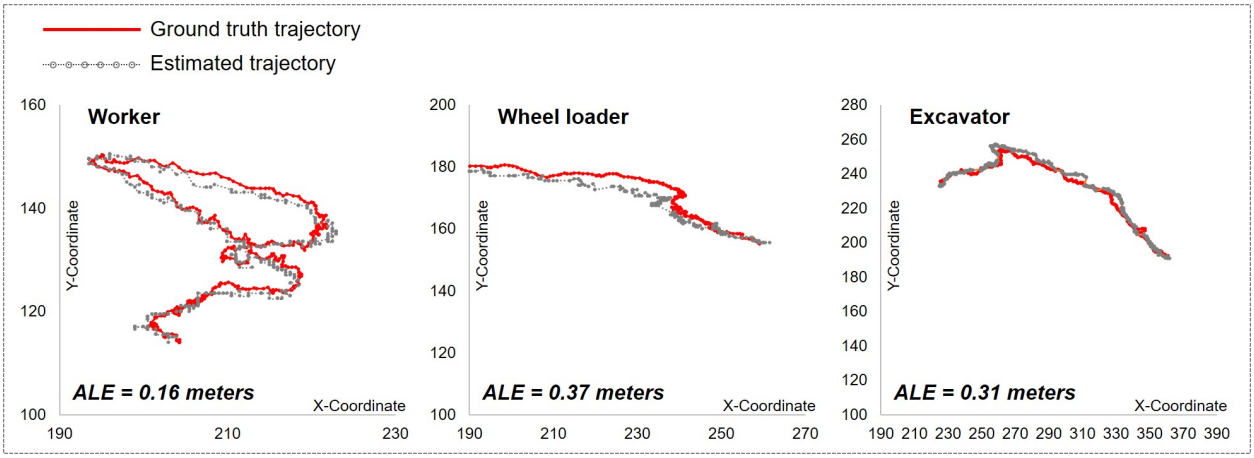

Figure 2.5 Result of Object Localization by YOLO-V3: Trajectories

2.6 Thrust #2: Image Rectification for Distance Measurement

While a camera maps 3D space onto a 2D image plane, projective distortion emerges, distorting original properties of a scene. Figure 2.6 provides a detailed example of the projective distortion. In the left-side image, the two ellipses are actually circles having same properties (i.e., diameter = 27.4 m), and also the tetragonal object is a square (i.e., width = height = 2.89 m). As such, measured proximity on a 2D image must be distorted and unreliable. While previous studies have focused on recovering depth information on a 2D image, this research approaches this problem by focusing on the removal of this projective distortion. The key insight is that the 3D distance between two objects placed on the same plane can be measured even with a 2D image if the projective distortion can be successfully rectified (Figure 2.6). That is, instead of measuring the depth of points, this research homogenizes the 3rd coordinates of points, thereby making distance measuring possible on a 2D image with a minimum computation. Along this way, this method leverages a reference object whose dimension is already known (e.g., a column foundation). This reference provides a geometric cue to estimate the homography between a distorted and rectified image as well as allows to measure the unique scene scale. After the rectification, the proximity can be measured in a metric unit, and the contact-driven hazard can be visualized considering the unique scene scale.

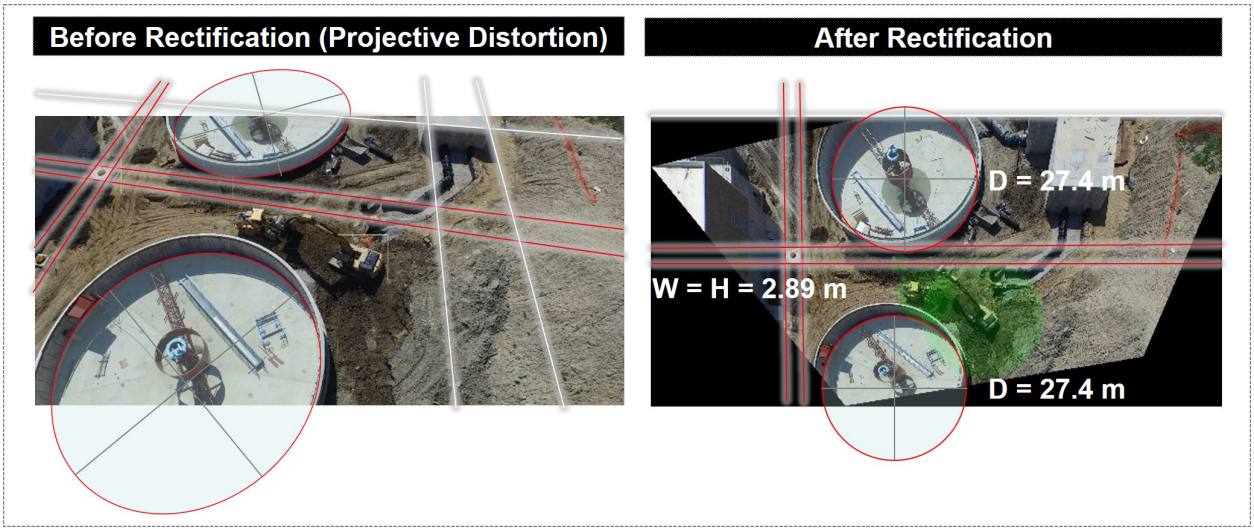

Figure 2.6 Projective Distortion: Before and After Rectification

2.6.1 Method description

The proposed method consists of the following six steps: (i) edge detection; (ii) line fitting; (iii) rectification; (iv) proximity measurement; (v) outlier filtering; and (iv) hazard visualization (Figure 2.7). The detail explanations are stated as follows.

- Edge detection: The Canny operator is used to detect the edges of the reference object. Because the bounding box of the reference can be given from the fine-tuned object detector, it can be applied only to the inside of the box so that the unnecessary edges irrelevant to the reference object can be filtered out. Firstly, the Gaussian filter (size = 7 x 7, sigma = 1) is applied to remove noises on the input image. Then, the Sobel operator generates the edge map with its magnitude and direction. Subsequently, the non-maximum suppression refines candidate edges to have the minimum thickness. Lastly, the hysteresis thresholding (i.e., high threshold = 0.6 and low threshold = 0.24) filters out the false positive edges. Accordingly, delicate (i.e., one-pixel thickness) and accurate edges can be detected, which are used as samples for fitting the reference object's contours in the next step.

- Line fitting: Using the detected edges, the contours of the reference object can be inferred. Among several line fitting methods (e.g., HOUGH transform), this method adopts the RANdom Sample Consensus (RANSAC) that discounts outliers for robust operation. Through the RANSAC line fitting, the best lines passing through detected edges are inferred as contours of the reference object. Firstly, two points are sampled at random. Then the line passing through them is drawn with its inline zone. In sequence, the number of inliers is counted. By iterating this, the best line having the largest number of inliers is saved as a contour. In this method, the threshold value of the one-pixel distance is used for determining inlier boundary and the model is iterated 2,000 times for fitting one contour. Through the RANSAC, the four contours of the reference object are inferred, and in turn, the four anchor points (i.e., the crossing point of two contours) can be detected.

- Rectification: The way to rectify a distorted image starts from finding the geometric transformation matrix (i.e., homography) that links the distorted dimension to the corresponding ground truth. By matching the estimated location of the four anchor points and that of the ground truth, the linear equation, i.e., Equation 2.4, is established, which can be solved by direct linear transformation (DLT) algorithm using singular value decomposition (SVD). Once the 3x3 homography is found, the whole frame can be rectified by applying this homography to every single pixel over a frame (Equation 2.5).

- Proximity measurement: After removing the projective distortion, the proximity between a worker and equipment can be estimated by calculating the Euclidean distance between them. In doing so, the pixel distance is converted to the metric unit, considering the scene scale known from the reference object's dimension (Equation 2.6).

- Outlier filtering: The misdetection of an anchor point will deteriorate the overall process, resulting in irregular outliers of proximity. An outlier filter is therefore embedded to automatically detect and offset potential outliers. This filter tracks the mean value of previous two estimations and determines whether the current one is an outlier or not by inlier thresholding (i.e., inlier buffer=50 pixel distances). Once an outlier is detected, the filter replaces it with the mean value of the previous two estimations.

- Visualization: additionally, the contact-driven hazard around robot (or equipment) is visualized with a user-adjustable diameter. This research uses the action radius of equipment as a default value for the diameter of a contact-driven hazard.

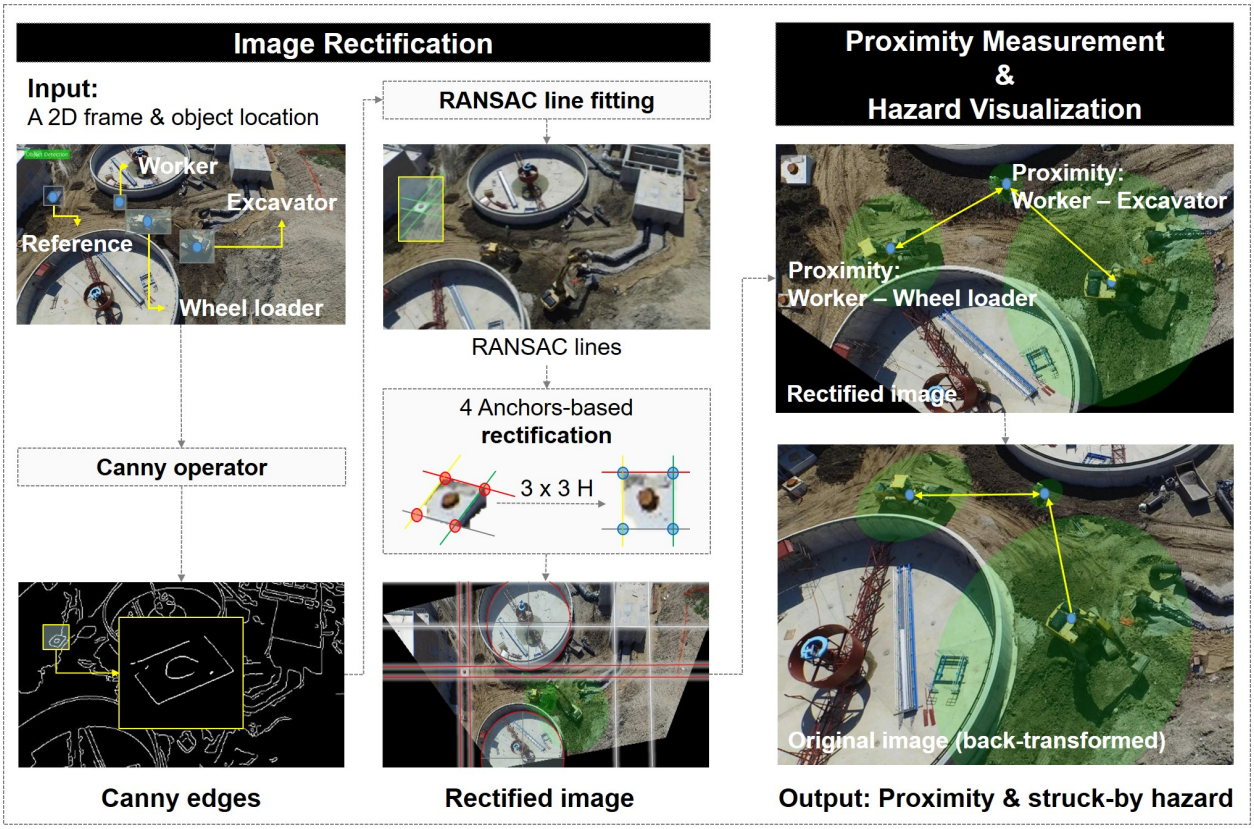

Figure 2.7 Overall Process of Automated Image Rectification

$$\begin{bmatrix} x_1 & x_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 \\ & & & \vdots & & & & \\ x_4 & x_4 & 1 & 0 & 0 & 0 & -x'_4x_4 & -x'_4y_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y'_4x_4 & -y'_4y_4 \end{bmatrix} * \begin{bmatrix} h_{11} \\ h_{12} \\ \vdots \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad \text{Equation 2.4}$$

Note: ($x_{(1\sim4)}$, $y_{(1\sim4)}$) stand for the estimated locations of anchor points; ($x'_{(1\sim4)}$, $y'_{(1\sim4)}$) stand for the ground truth location of anchor points; and $h_{(11\sim32)}$ stand for the elements of the 3x3 homography (h_{33} is always 1).

$$W \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \text{Equation 2.5}$$

Note: (X,Y) stands for the rectified coordinates of an original pixel; (x,y) stands for the coordinates of an original pixel; and W stands for a scale factor

$$Proximity_{meter} = \frac{Reference_{meter}}{Reference_{pixel}} * Proximity_{pixel} \quad \text{Equation 2.6}$$

Note: $Reference_{meter}$ stands for the ground truth width of the reference (unit: meter); and $Reference_{pixel}$ stands for the estimated width of the reference on the rectified image (unit: pixel)

2.6.2 Test result

A lab-scale test was conducted to evaluate the effect of rectification in measuring distance (i.e., proximity). Figure 2.8 illustrates the test settings. The 8x8 square checkerboard (width=height=25 cm) was used to describe a real ground plane (width=height=25 meter) with 1:100 scale. The left top corner was selected as a worker's location and the others as possible locations of equipment, from which the ground truths for the 48 proximities were established (Table 2.4). An aerial video was filmed using a mobile cell phone, by taking UAV-like motion (i.e., varying location and orientation), as if the video was recorded by a camera-mounted UAV (Figure 2.8).

This test measured the proximities both on original and rectified images, and compared them with pre-defined ground truth proximities. The overall accuracy (i.e., for before and after rectification) was determined by the mean absolute percentage error (MAPE) (Equation 2.7).

$$\text{Accuracy} = 100\% - \frac{1}{n} * \sum_{i=1}^{n} \left| P_g - P_e \right| / P_g * 100 \qquad \text{Equation 2.7}$$

Note: n stands for the total number of targets (i.e., 48); P_g stands for ground truth proximity; and P_e stands for estimated proximity.

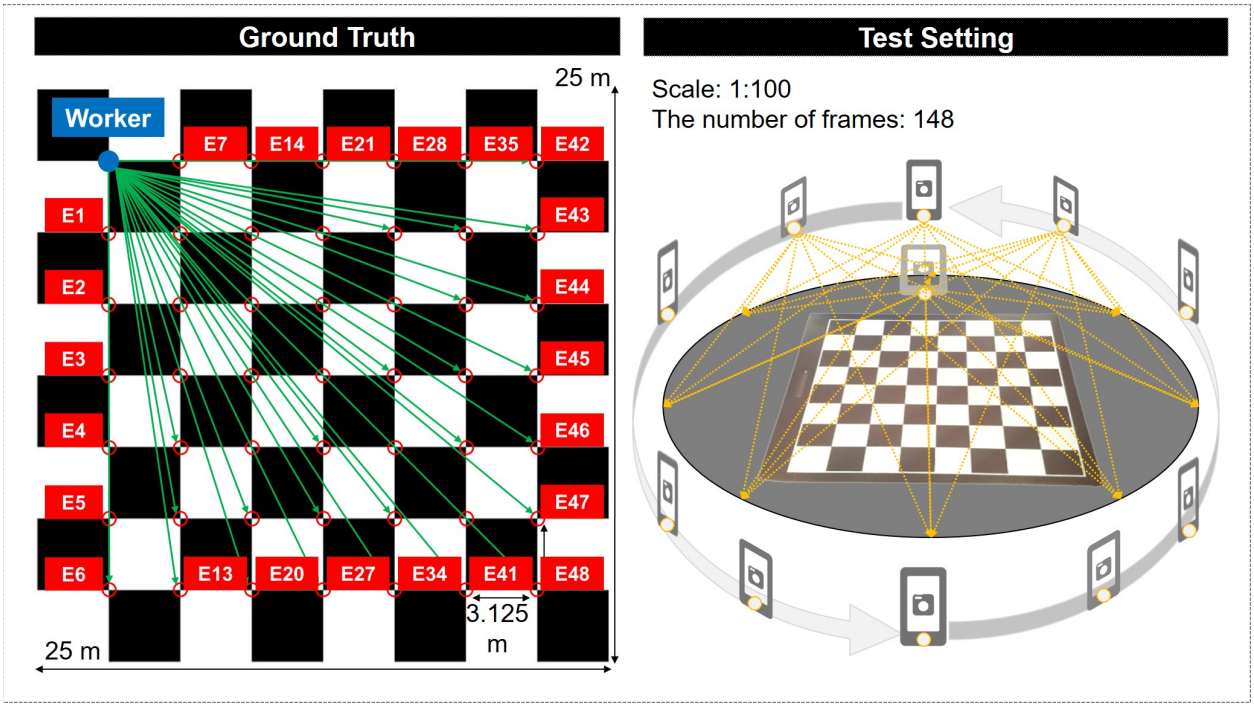

Figure 2.8 Rectification Test: Ground Truth vs. Test setting

As the result, it was shown that the average accuracy of proximity after the rectification was more than 97% (Table 2.5), which outperforms the original accuracy by 3.93 points (i.e., before=93.51%, Table 2.4). Furthermore, it was revealed that the effect of rectification is to be greater when a higher extent of projective distortion exists on an image. For example, in the case of the 110th frame of a diagonal viewpoint, the rectification could improve the accuracy by 25 points (i.e., before=68.32% and after=93.33%) (Figure 2.9). Given the fact that the extent of the distortion is far more serious in usual UAV-captured videos (Figure 2.6), the effect of rectification is expected to be greater than this lab scale test.

Table 2.4 Proximity Accuracy (Before Rectification)

Target #	Ground truth (Unit: meter)	Proximity after rectification (below row = frame #)						
		1	2	3	……	146	147	148
1	3.13	3.13	3.13	3.13	……	3.13	3.13	3.13
2	6.25	6.24	6.24	6.24	……	9.16	9.16	9.12
3	9.38	9.36	9.36	9.36	……	12.07	12.05	11.99
4	12.50	12.49	12.48	12.48	……	14.95	14.91	14.80
5	15.63	15.67	15.66	15.65	……	14.95	14.91	14.80
				.				
				.				
				.				
				.				
44	19.76	19.78	19.76	19.73	……	16.47	16.40	16.28
45	20.96	20.99	20.97	20.94	……	17.61	17.52	17.37
46	22.53	22.61	22.60	22.56	……	19.01	18.90	18.73
47	24.41	24.59	24.57	24.53	……	20.63	20.50	20.29
48	26.52	26.84	26.82	26.78	……	20.63	20.50	20.29
100% - MAPE		99.71	99.72	99.70	……	86.27	86.00	85.61
		Overall accuracy (%)						**93.51%**

Table 2.5 Proximity Accuracy (After Rectification)

Target #	Ground truth (Unit: meter)	Proximity after rectification (below row = frame #)						
		1	2	3	……	146	147	148
1	3.13	3.13	3.13	3.13	……	3.13	3.13	3.13
2	6.25	6.24	6.24	6.23	……	6.24	6.24	6.24
3	9.38	9.36	9.34	9.31	……	9.35	9.33	9.34
4	12.50	12.50	12.46	12.40	……	12.47	12.43	12.46
5	15.63	15.69	15.63	15.53	……	15.63	15.55	15.60
				.				
				.				
				.				
				.				
44	19.76	19.91	19.93	19.84	……	20.43	20.15	20.23
45	20.96	21.11	21.12	21.00	……	21.65	21.33	21.43
46	22.53	22.74	22.73	22.56	……	23.29	22.91	23.05
47	24.41	24.73	24.70	24.47	……	25.31	24.86	25.03
48	26.52	27.01	26.96	26.66	……	27.61	27.07	27.28
100% - MAPE		99.58	99.00	99.55	……	92.87	91.60	92.87
		Overall accuracy (%)						**97.43%**

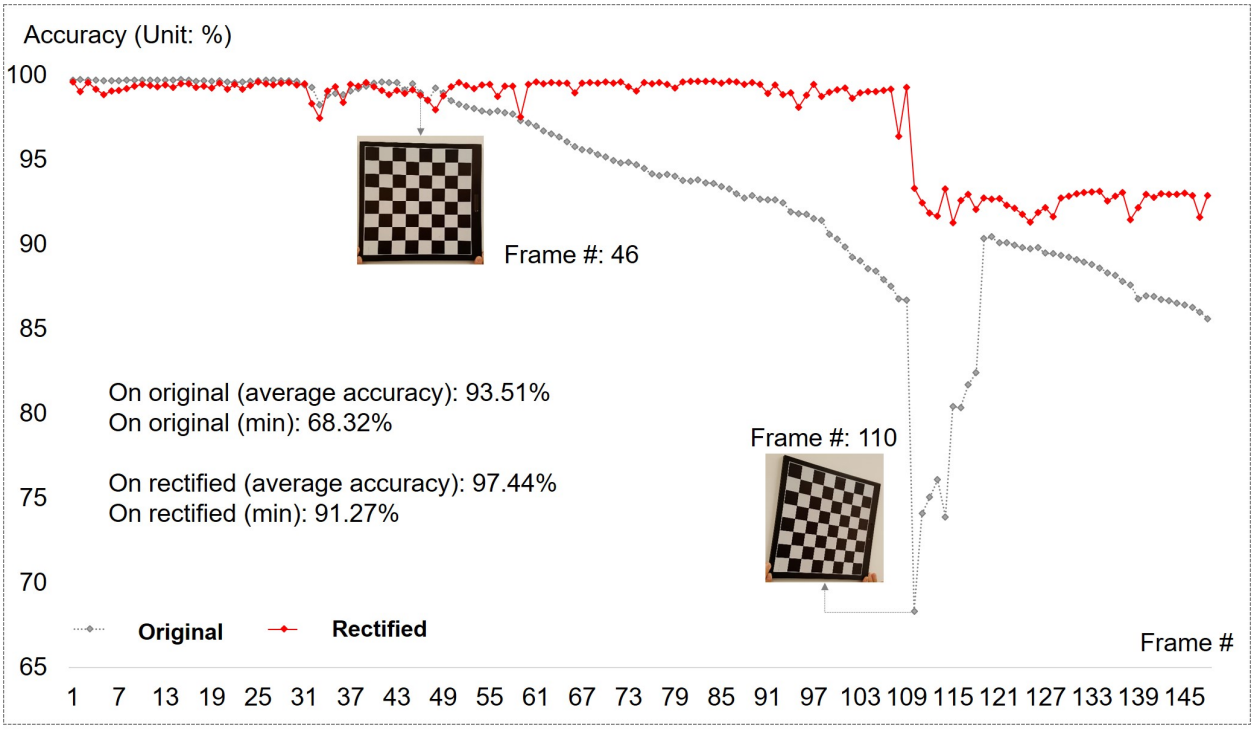

Figure 2.9 Proximity Accuracy: Before vs. After Rectification

2.7 Test on Real-Site Aerial Videos

To evaluate the proposed method's accuracy in real-world application, this research conducted two tests on real-site aerial videos. The first tested the ability for mobile construction entities to work a normal operation whereas the second test targeted stationary entities in a controlled environment.

2.7.1 Test on mobile construction entities

Table 2.6 provides the overview of the first test. The video was filmed at a real construction site by a camera mounted-UAV. It is comprised of 10,614 consecutive frames. The 398 frames capturing worker-equipment interactions were sampled for this test. In this work, the proximity between a worker and two pieces of equipment (i.e., wheel loader and excavator) were analyzed (Figure 2.10).

Table 2.6 Overview of the Test for Mobile Entities

Categories		Description
The # of total frames		10,614
The # of frames analyzed		398
Resolution		3840 x 2140
Target's action radius	A worker	2 meters
	A wheel loader	5.8 meters
	An excavator	12.1 meters
Reference object	A quadrate concrete footing	Dimension (meters): width=height=2.89
Evaluation metrics		ADE and MAPE

The primary challenge of this test was to secure a comparison benchmark. While it would have been ideal to directly measure ground truth proximity on the site while filming the video, it was a challenge to measure the proximity on the field without interrupting the site operations, while also facing additional barriers to implementation (e.g., safety issues). As an alternative, we used entities' location information, which we annotated manually, and applied a statistical inference process to secure a reasonable substitute for the ground truth proximity. Once correct locations for the two targets were given, errorless rectification allowed for calculating the ground truth proximity between them. In the real scene application, however, the rectification could be influenced by noises, which can result in a ground truth estimation dispersed with outliers. As shown in Figure 2.11, this raw estimation (i.e., each point) itself cannot be reliable as it contains a wide scope of errors and ignores continuity of a proximity. However, the obvious trend line exists in there, which can be a valid comparison benchmark, once a reasonable inference process is given. The following steps were applied to attain this end (Figure 2.11): (i) removing outliers by thresholding; (ii) fitting baseline (i.e., dotted line); (iii) removing additional outliers from the baseline with 1.0 standard deviation; and (iv) fitting the final trend line (i.e., solid line). In this test, 9^{th}-order polynomial model was used for fitting the baseline and the final trend line, considering the proximity pattern of given test dataset.

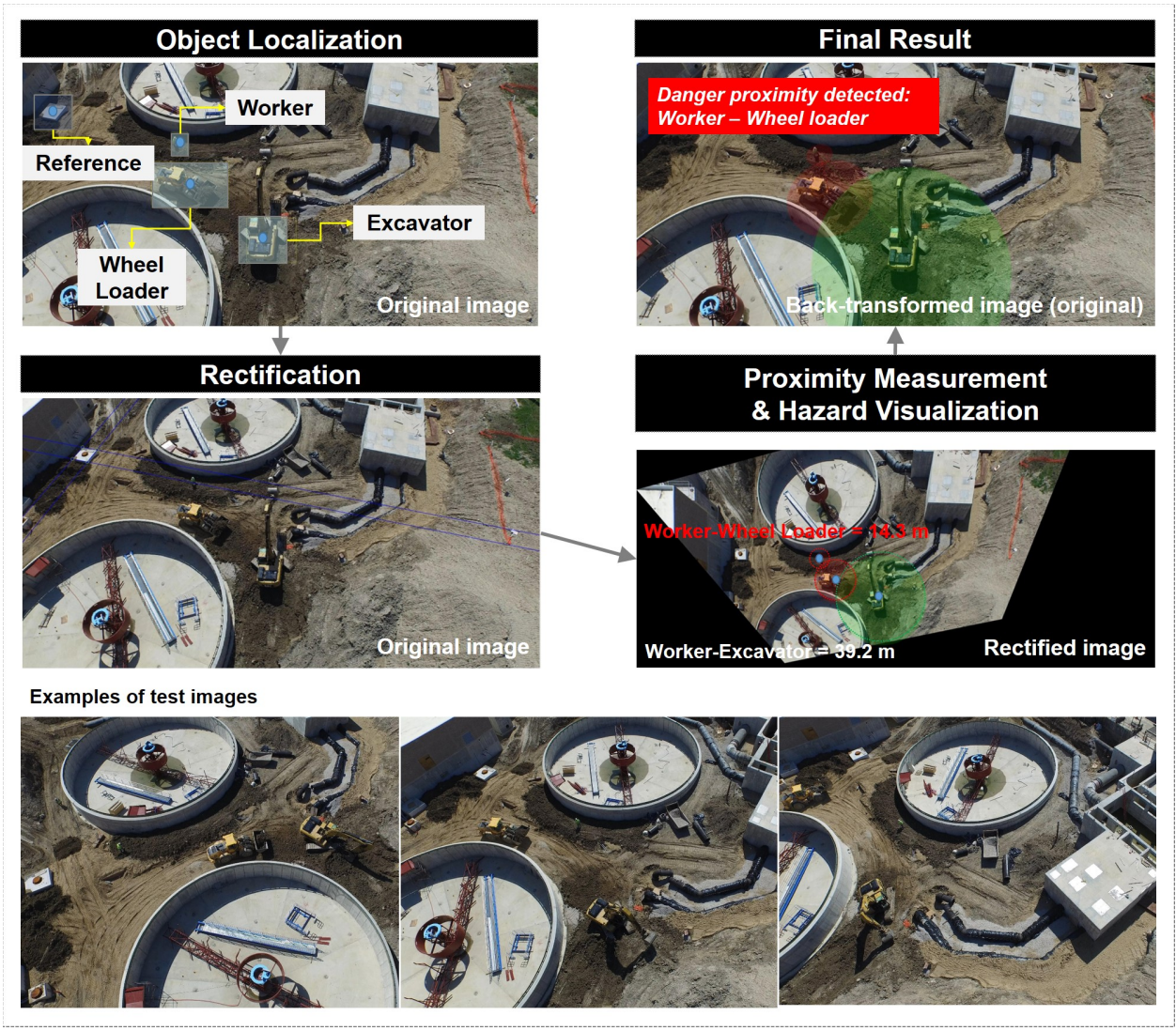

Figure 2.10 Test on Mobile Construction Entities: Operational Procedure

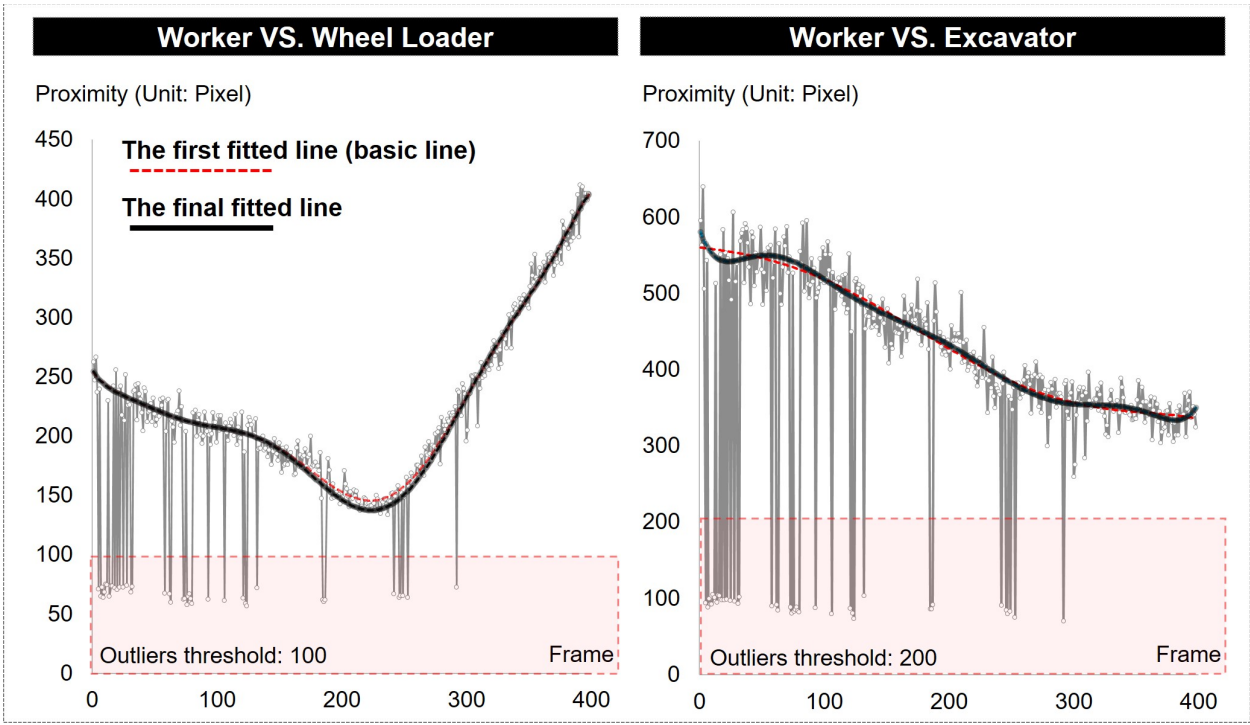

Figure 2.11 Inference on Comparison Benchmark

As shown in Figure 2.12, the proximity estimate was compared to the comparison benchmark. As an evaluation metric for accuracy, the mean absolute distance error (MADE) was used (Equation 2.8) along with the corresponding MAPE (Equation 2.9). It was shown that the estimation was close to the benchmark proximity in both cases (i.e., worker-wheel loader and worker-excavator) with the acceptable MAPE: 3.72% and 4.85%, respectively. The MADE for worker—wheel loader was 0.33 meters and that for worker—excavator was 0.89 meters. Moreover, the proposed method showed unbiased performance with having evenly spread distance errors (i.e., residuals) around median values (i.e., 0.01 meters and -0.16 meters, respectably).

$$\text{MADE} = \frac{1}{n} * \sum_{i=1}^{n} |P_b - P_e| \qquad \textit{Equation 2.8}$$

$$\text{MAPE} = \frac{1}{n} * \sum_{i=1}^{n} |P_b - P_e| / P_b * 100 \qquad \textit{Equation 2.9}$$

Note: n stands for the number of frames (i.e., 398); P_b stands for benchmark proximity; and P_e stands for estimated proximity.

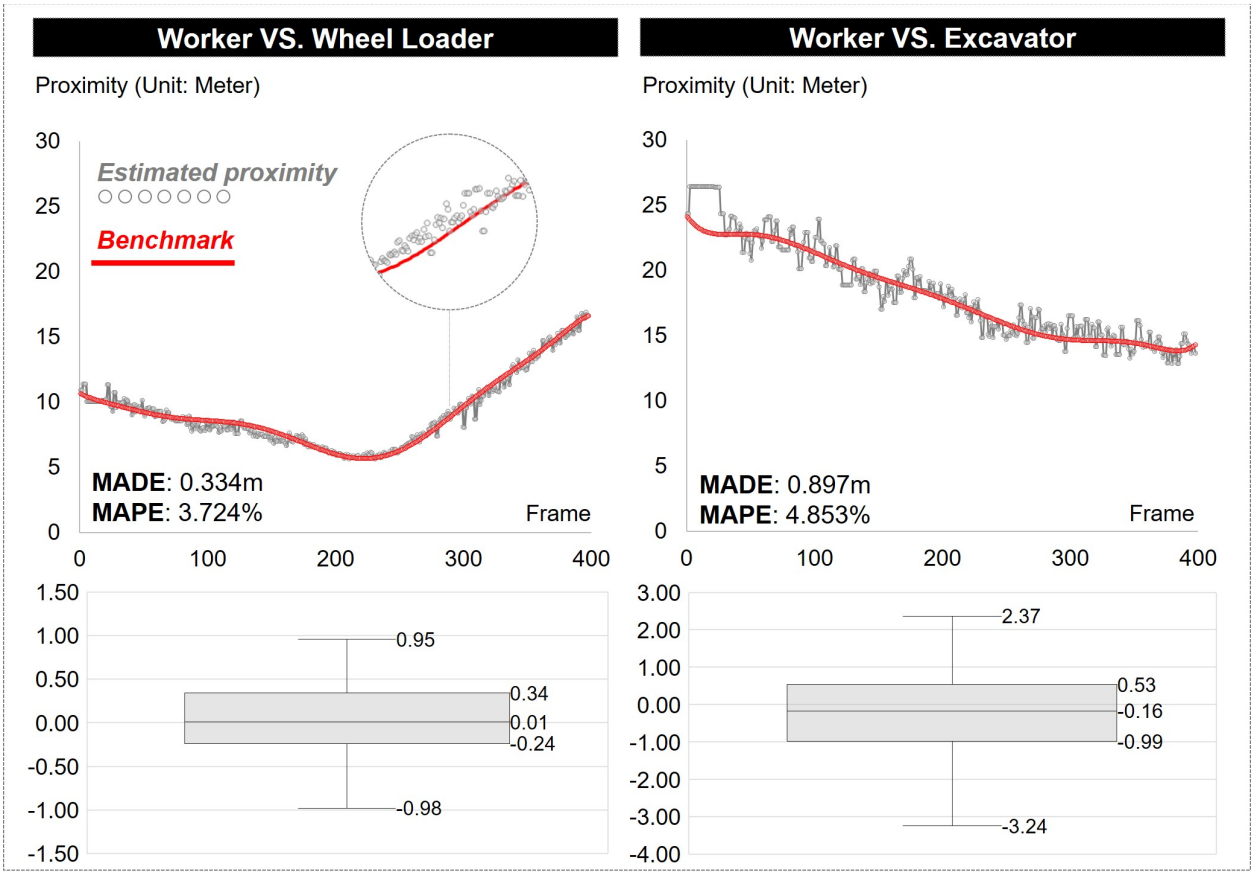

Figure 2.12 Test Result for Mobile Entities: Estimation vs Comparison Benchmark

2.7.2 Test on stationary construction entities

The details of the second test are summarized in Table 2.7 and Figure 2.13. An aerial video was filmed at a real construction site using a mobile cell phone, by taking UAV-like motion. Unlike the previous work, this test fixed the locations of targets to secure the ground truth proximity. In this test, the proposed method estimated the proximity between a stationary worker and an excavator (Figure 2.13). And the estimate was compared to the pre-defined ground truth proximity. Two cases of ground truth proximity were analyzed in this test: (i) case #1: 15 meters and (ii) case #2: 20 meters (Figure 2.13).

Table 2.7 Overview of the Test for Stationary Entities

Categories		Description
Resolution		1920 x 1080
Ground truth	Case #1	15 meters
	Case #2	20 meters
The # of frames analyzed	Case #1	50 frames
	Case #2	50 frames
Target	A worker	Stationary
	An excavator	Stationary
Reference object	A tetragonal concrete footing	Dimension (meters): 2.4-2.6-0.6-2.6-0.8
Evaluation metrics		ADE and MAPE

The proximity estimate was compared to the ground truth. It turned out that the estimation was close to the ground truth in both cases (i.e., 15 meters and 20 meters) with the acceptable MAPE: (i) case #1: 4.214% and (ii) case #2: 4.462% (Figure 2.14). The MADE for the case #1 was 0.632 meters and that for the second case was 0.892 meters (Figure 2.14).

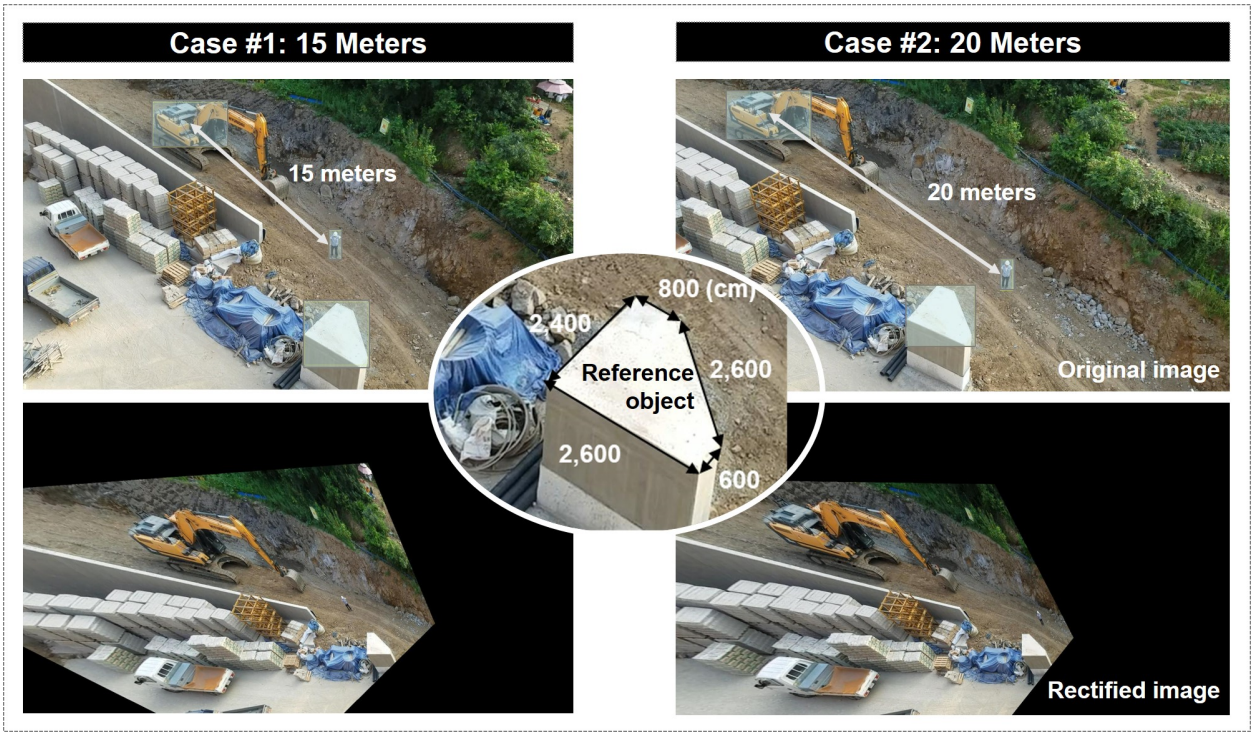

Figure 2.13 Test on Stationary Entities: Operational Procedure

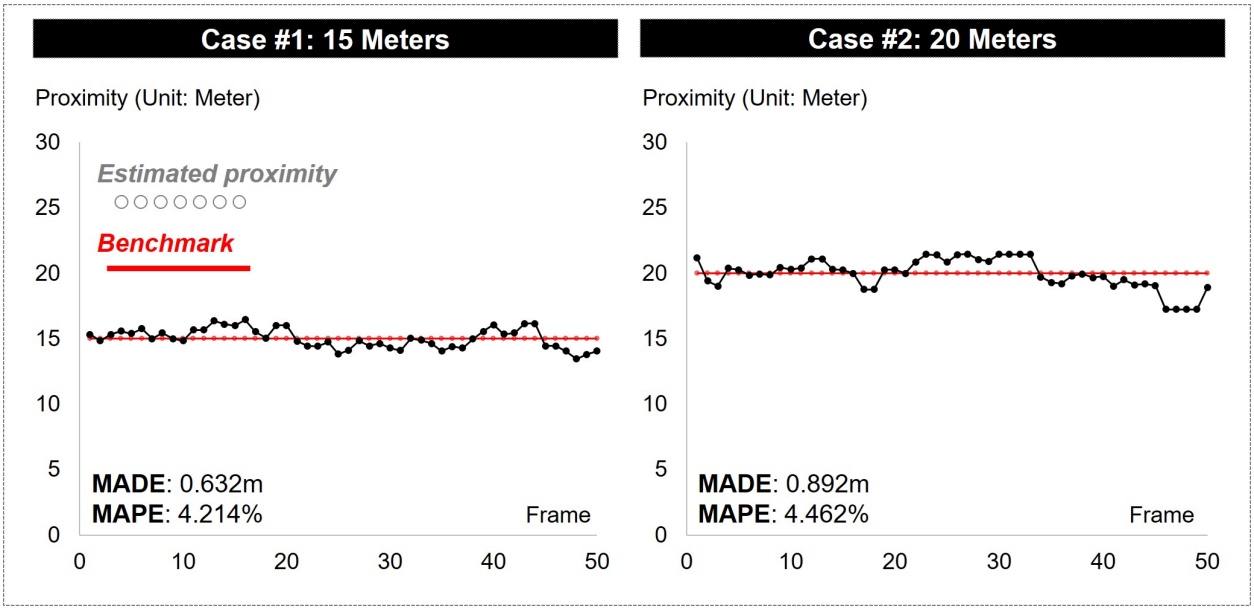

Figure 2.14 Test Result for Stationary Entities: Estimation vs Ground Truth

2.8 Discussion on Test Results

In the first test to target mobile construction entities, the fine-tuned detector (i.e., YOLO-V3) showed robust localization performance; the localization error (Equation 2.3) for the three construction entities (i.e., a worker, a wheel loader, and an excavator) could be held around 0.3 meters even under viewpoint, scale, and illumination variations occurring in the test videos. In achieving the invariant performance were two primary contributories: (i) transfer learning; and (ii) fine-tuning with the data having a wide range of variations. First, balancing between the model capacity and the amount of training data is critical in avoiding overfitting. However, the amount of data collected in this research (i.e., 4,512 images) was not ideal for training the original YOLO-V3 architecture to have deep layers (i.e., total 106 layers) from scratch. This research, therefore, elected transfer learning. To be more specific, I took the YOLO-V3 network pre-trained with ImageNet benchmark dataset and used its weights as the starting point of fine-tuning. Naturally, network modifications were made for fitting the original architecture to our dataset (i.e., adjustment of the size of the final feature tensors). By starting from pre-validated weights, the network could achieve well-balanced training without overfitting, thereby making it possible to have an equivalently robust performance on both the training and test dataset. Second, fine-tuning with data involving a wide range of variations helped to enhance the invariant localization

capability. This research primarily used images extracted from the videos captured in various construction sites, which covered a wide range of variations regarding illumination, viewpoint, and scale. Fine-tuning with the variable data helped to optimize parameters, e.g., coefficients of convolution kernels, to be invariant to such variations. The parameters could construct consistent feature tensors in successive frames, which in turn led to the robust localization results.

With the localization result, the image rectification method could lead to a reliable proximity measurement between the three entities, successfully removing the projective distortion. First of all, the anchor-points detection using the Canny operator and the RANSAC line fitting was hardly affected by the viewpoint, scale, and illumination variations with advantages of non-maximum suppression and hysteresis thresholding. Given the precise locations of the anchor-points, the rectification method could solve the unique solution for the geometric transformation matrix toward the undistorted original scene and thus could get reliable proximity estimates. On the other hand, the rectification could not be successful at times (i.e., 37/398, in the test for mobile construction) due to aggregates of noise pixels (e.g., sands covering the reference objects). However, all outliers of the estimated proximity resulted from the rectification failures could be successfully detected and refined by the outlier filter. As the result, the proposed method could achieve a promising accuracy of the proximity estimate (i.e., worker—wheel loader: 0.33m MADE and 3.72% MAPE, worker—excavator: 0.89m MADE and 4.85% MAPE). As in the test with a virtual image (Figure 2.15), the use of unsoiled reference object that has clear contour lines and distinctive color to surroundings would be the one simple but powerful solution for this problem. Furthermore, designing a new filter that can automatically remove aggregates of noise pixels having non-linear patterns could also be an effective solution to reduce the chance of a mis-rectification. In real-world applications, specifically, when detecting contact-driven hazards, this minor amount of error would be offset by adding an extra buffer (e.g., 1 meter) to the action radius of equipment.

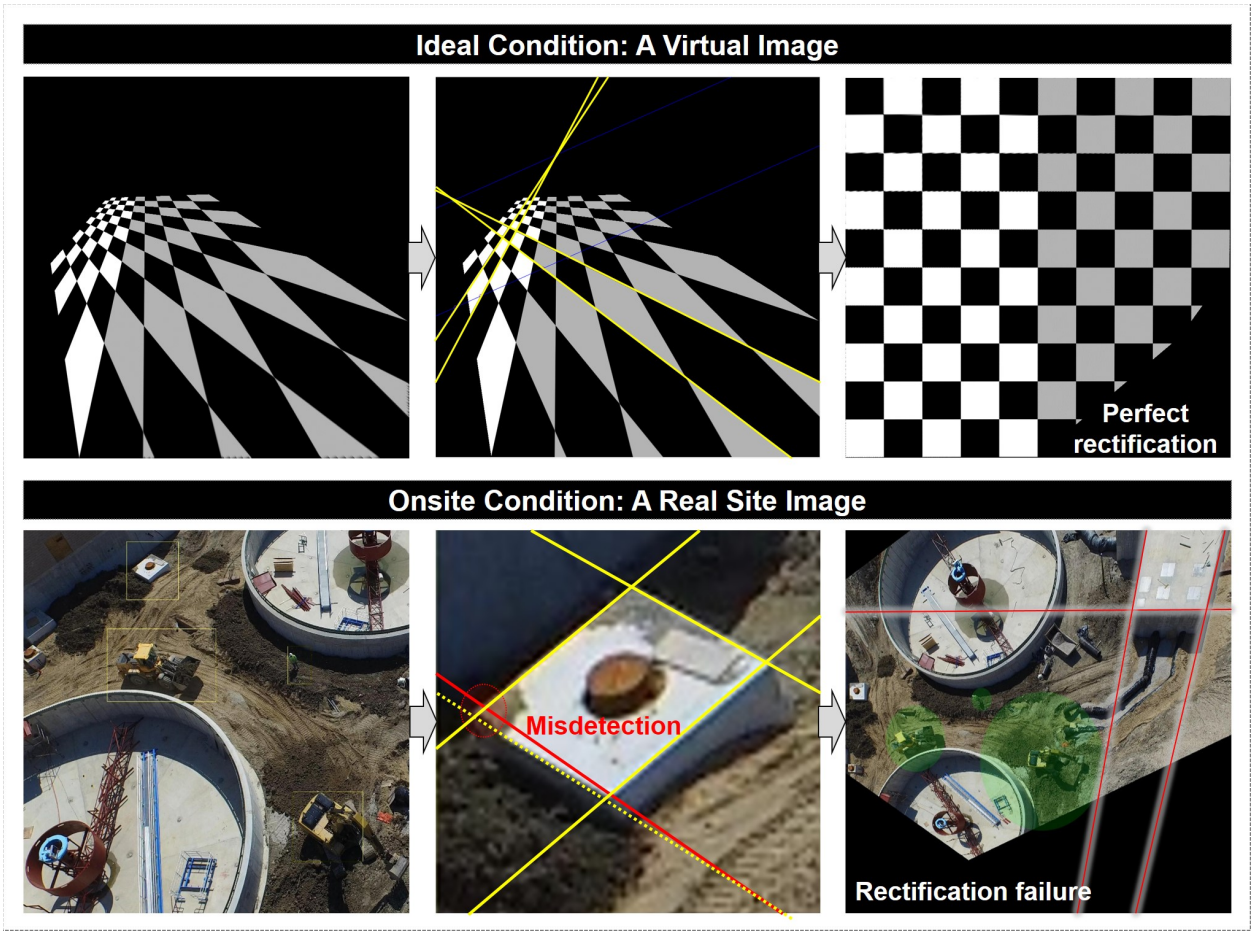

Figure 2.15 Rectification Performance: Virtual Condition vs. Onsite Condition

Following the first test, I conducted an additional test focusing on stationary targets (i.e., an excavator and a worker). This test is designed to compare the proximity estimates from our method with the ground truth proximity directly measured on the site in order to validate the proposed method more convincingly. Consequently, the proposed method showed promising accuracy in the second test as well. The MADEs for both cases (i.e., 15 meters and 20 meters proximity) was less than one meter and corresponding MAPEs were around 4%. The results clearly show the validity of the proposed method.

2.9 Conclusions

This study aimed to address the first agenda of the robotic hazard detection roadmap, which is proximity monitoring between workers and activated (mobile) robots. To achieve less-occluded, real-time, and accurate proximity monitoring, I specifically leveraged camera-mounted UAVs as imaging devices and developed a real-time visual proximity monitoring method leveraging DNN-powered computer vision and image processing techniques.

A DNN for object detection, i.e., YOLO-V3, was applied to the robust and fast localization of construction entities. In addition, an image rectification method that allows for measuring actual proximity on a 2D image was developed. When operated together, these methods can consistently monitor proximity between construction entities in a fully automated way. Tests on real-site aerial videos showed promising performance of the proposed method; the MADEs were less than 0.9 meters and the corresponding MAPEs were around 4%. However, there still remains plenty of room for improvement: (i) improving the generalization capability of the fine-tuned network and (ii) improving the computational efficiency of the rectification method. With such critical refinement, the proposed method can serve as an effective proximity monitoring method in the conclusive hazard detection roadmap.

CHAPTER 3

Proximity Prediction using a Conditional Generative Adversarial Network[2]

3.1 Introduction

Following-up on the prior study for proximity monitoring, this study further investigated the potential of proximity prediction in future time-steps. Prediction can be far more important and effective for contact-driven accident prevention (Kim et al. 2019c)—principally because the sooner robot and worker are informed of their proximity to each other, the more likely they are to avoid a potential collision. Proximity monitoring is essential for robotic hazard detection; however, it may not be as effective in highly impending situations. In a dynamic and unstructured construction site, contact-driven accidents occur spontaneously in unexpected ways. In such an impending situation, monitoring proximity at the current time-step would not be effective because the near-sighted measure would not allow enough time for the involved robot (and equipment operator) and worker to take prompt evasive action. In this sense, to better prevent contact-driven accidents in co-robotic construction, hazard detection technology needs to be equipped with prediction functionality.

[2] This chapter is adopted from Kim, D., Lee, S., and Kamat, V.R. (2020) "Proximity Prediction of Mobile Objects to Prevent Contact-Driven Accidents in Co-Robotic Construction." *Journal of Computing in Civil Engineering*, 34(4), 04020022.

In this context, I updated the prior proximity monitoring method by adding an additional module for trajectory prediction. As with the prior study, this study leveraged a camera-mounted UAV to monitor associated entities (Figure 3.1). Inputting the UAV-captured imagery data, the updated method powered by DNNs for object detection (Figure 1-A) and trajectory prediction (Figure 1-B) performs proximity prediction (Figure 1-C) in a fully automated way.

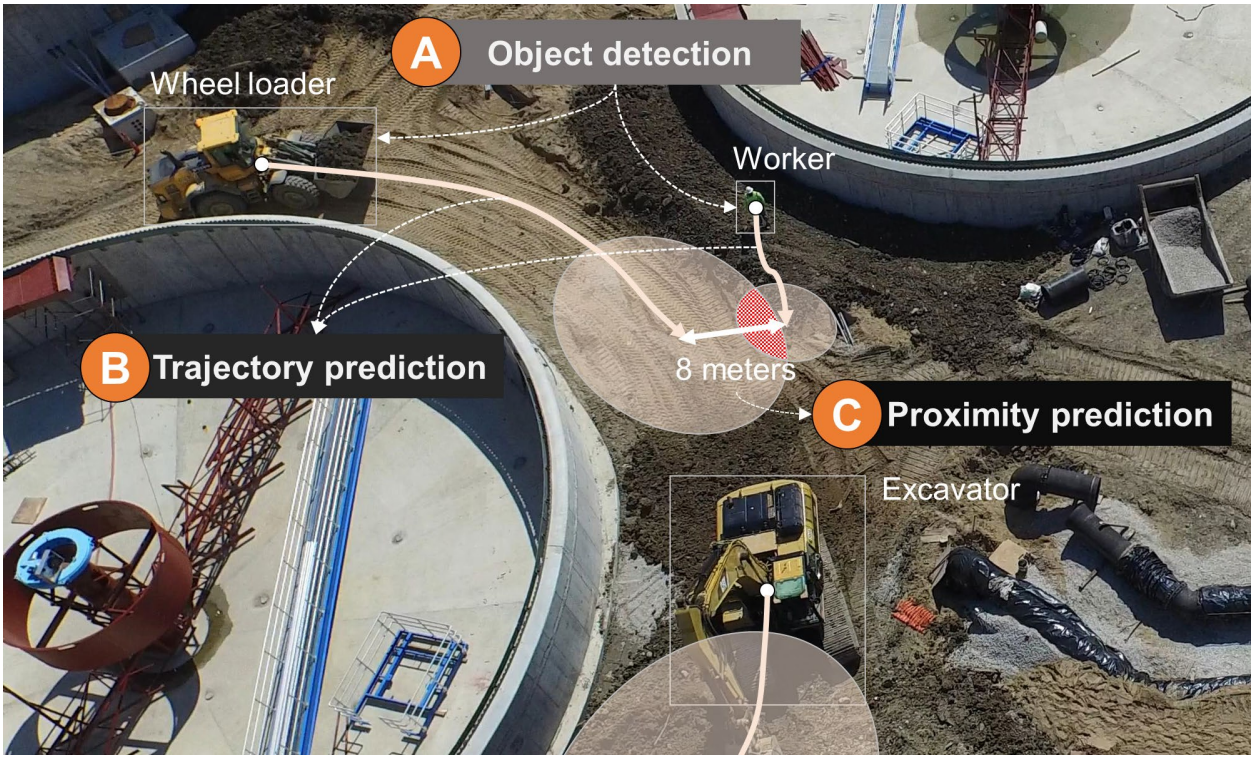

Figure 3.1 Proximity Prediction using a Camera-Mounted UAV and DNNs

The major contribution of this work is to enable predicting risks of impending collision, thereby making pro-active safety interventions possible. Specifically, proximity prediction would assist mobile robots' predictive path planning and rerouting. Also, via wearable devices (e.g., wrist band and smart safety glasses), this capability would enable providing an advance alert to workers, helping them to take timely evasive action. These pro-active interventions would effectively reduce the chances of impending collisions between mobile robots (or mobile equipment) and construction workers. Moreover, I applied a generative adversarial network (GAN) to trajectory prediction, which opens a new possibility of using GAN for construction applications.

3.2 DNN-based Framework for Proximity Prediction

The proximity prediction framework consists of two main modules: (i) a trajectory observation module that monitors targets' locations and records their past trajectories and (ii) a trajectory prediction module that predicts the target's future trajectories and estimates their future proximity. This section details each module's functionality and development process as well as presents its validation result.

3.2.1 Module 1: Trajectory observation

The first module monitors targets' locations and records their past trajectories, which are the primary input for trajectory prediction (Figure 3.2). This module first detects targets on a UAV-captured input image and estimates their center location as image coordinates (i.e., x-y pixel coordinates) using an object detection model based on YOLO-V3 (Figure 3.2(a)). In turn, this module rectifies the coordinates to the world coordinates through geometric transformation using a reference object since the image coordinates can neither reflect the true scene scale nor be accurate due to a projective distortion inherent on a 2D image captured by a UAV (Figure 3.2(b)). This module runs the object detection and the coordinate rectification at every input image, thereby continuing to update true-to-scale, distortion-free locations of targets. Based on the location information, it records the targets' past trajectories (from 3.96 seconds earlier to current, Figure 3.2(c)) and streams those to the second module for trajectory prediction.

The primary role of Module 1 is the trajectory observation of mobile construction objects but it can also conduct real-time proximity monitoring. In the Chapter 2, I demonstrated this module's performance on proximity monitoring—0.26 meters average displacement error (i.e., average of Euclidean distance between a target's ground truth and estimated positions) and 0.61 meters average proximity error (i.e., average of absolute difference between a pair of targets' ground truth proximity and estimated proximity). The details of Module 1's proximity monitoring performance can be found in the Chapter 1.

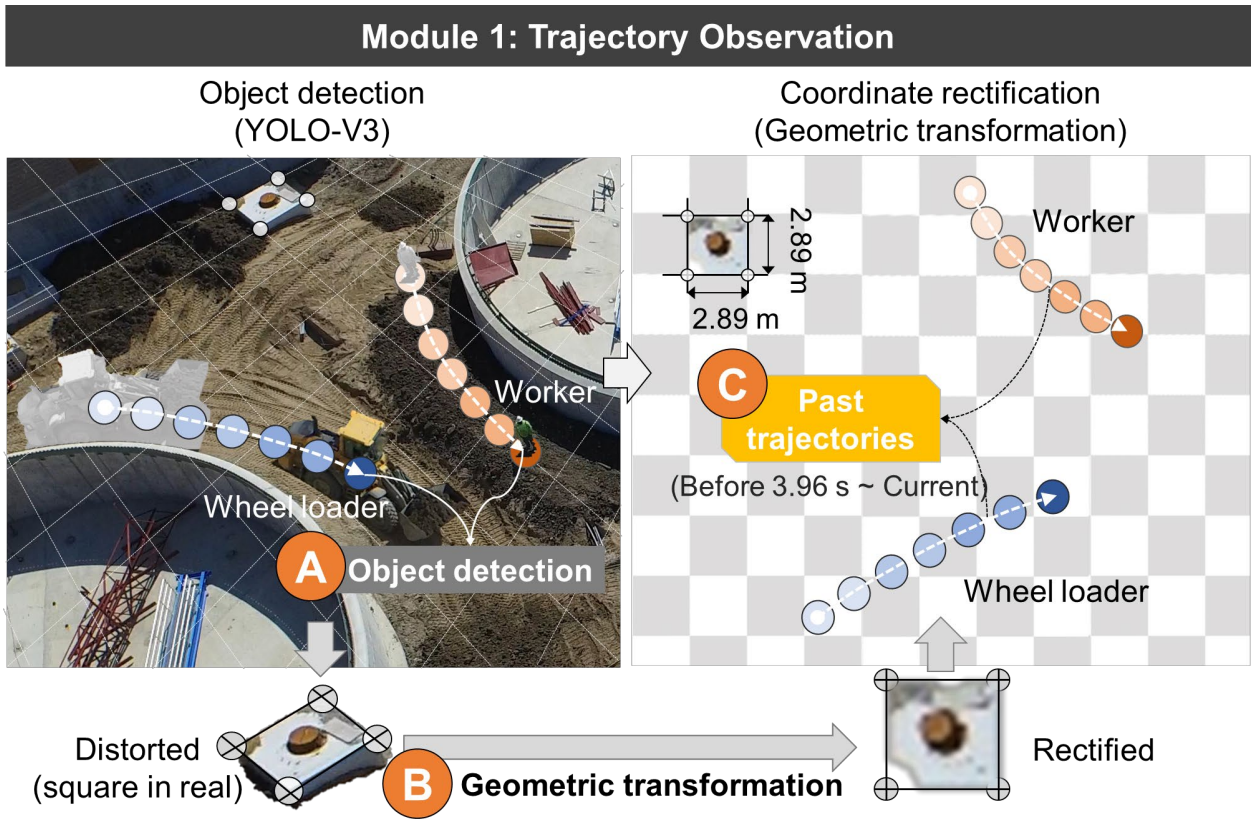

Figure 3.2 Module 1: Trajectory Observation via Object Detection and Coordinates Rectification

3.2.2 Module 2: Trajectory prediction

The second module (i.e., trajectory prediction) takes a set of target's past trajectories as input (from 3.96 seconds earlier to current, Figure 3.3(a)) and predicts their future trajectories for up to 5.28 seconds (Figure 3.3(b)), using a trajectory prediction model based on Social GAN (S-GAN, Gupta et al. 2018). The set of future trajectories informs where the targets will be located for the next 5.28 seconds at an interval of 0.66 seconds. Lastly, based on the targets' predicted locations, this module estimates the targets' proximity for the next 5.28 seconds—the proximity after 0.66, 1.32, 1.98, 2.64, 3.30, 3.96, 4.62, and 5.28 seconds (Figure 3.3(c)).

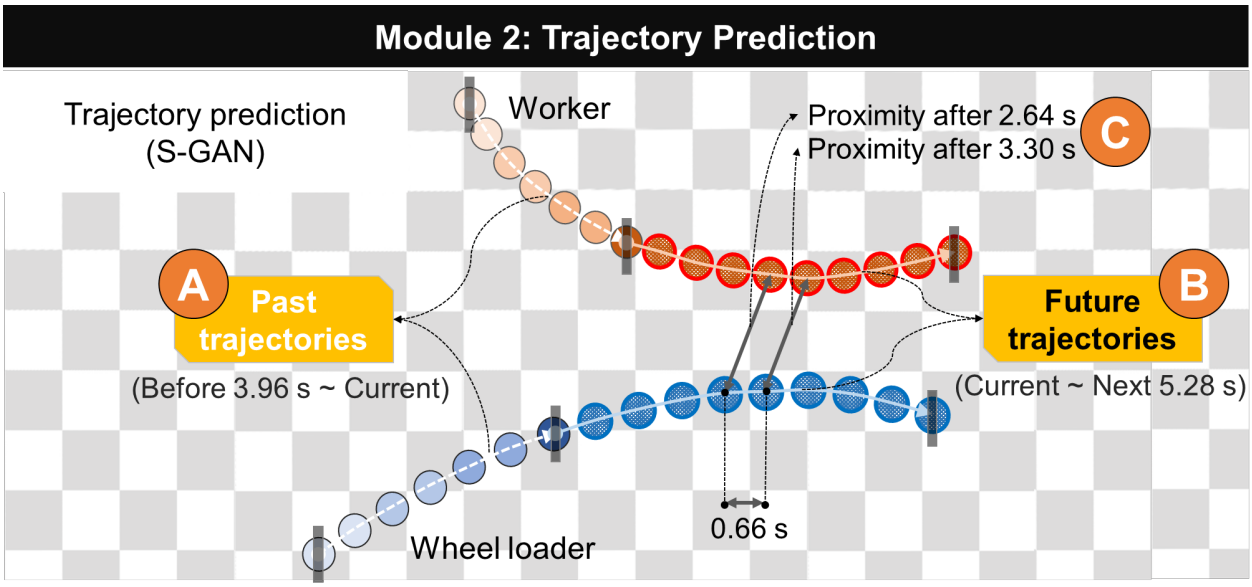

Figure 3.3 Module 2: Trajectory Prediction using S-GAN

Trajectory prediction studies have been dominated by data-driven learning approaches. This is basically because the movement of an entity (e.g., people) is so diverse and uncertain that it is extremely challenging to model through hand engineering. In an early stage, there are several studies to use hand-crafted features-based learning (Yamaguchi et al. 2011; Antonini et al. 2006; Helbing and Molnar 1995) or statistical learning such as polynomial regression (Rashid and Behzadan 2017), Gaussian process (Trautman et al. 2015; Tay and Laugier 2008), and hidden Markov model (Rashid and Behzadan 2017). However, many contemporary studies are motivated to use a DNN, following the trajectory of many other data-driven studies. In recent years, several DNN architectures for trajectory prediction have been released: for example, there are social long short-term memory (S-LSTM, Alahi et al. 2016), crowd interaction DNN (Xu et al. 2018), interaction aware DNN (Pfeiffer et al. 2018), and S-GAN (Gupta et al. 2018). Of these, the S-GAN, incorporating several distinctive features, demonstrated a state-of-the-art performance over others (Gupta et al. 2018). It enables a model to learn social behavior (e.g., collision avoidance) as well as an entity's moving pattern by integrating an LSTM encoder-decoder and a social pooling layer (Gupta et al. 2018). By realizing GAN architecture (i.e., coupling discriminator to generator) and adversarial training, it enhances the capability to learn complicated distributions of mobile objects' trajectories and improves reliability of prediction output. For this reason, this study applied the S-GAN and developed a trajectory prediction model through transfer learning.

3.2.2.1 Network architecture of S-GAN

The S-GAN has two main components: (i) generator that predicts targets' future trajectories (Figure 3.4(a)) and (ii) discriminator that inspects the quality of the predictions (Figure 3.4(b)).

- Generator (Figure 3.4(a)): the generator takes past trajectories of targets as input and predicts their future trajectories through network integrating social pooling layer into the middle of LSTM encoder-decoder. The generator first converts the input trajectories to fixed-length vectors via multilayer perceptron (MLP, Figure 3.4(aa)) and feeds it to LSTM units of encoder (figure 3.4(ab)). The LSTM units then encode the targets' movement patterns individually and forward the encoded features to social pooling layer which infers the targets' social interactions and generates pooled tensor for each target (Figure 3.4(ac)). Lastly, the decoder interprets the interconnected hidden state of input trajectories with multiple LSTM units and generates socially plausible future trajectories of the targets (Figure 3.4(ad)). Here, the decoder initializes itself with input trajectories so that it can generate future trajectories that better conform to the past ones.
- Discriminator (Figure 3.4(b)): the discriminator inspects the predicted trajectories' quality and conformity to the past trajectories. It takes both of past and future trajectories together as input and encodes their conformity features through LSTM units (Figure 3.4(ba)). In turn, it calculates the predicted trajectories' conformity score via MLP (Figure 3.4(bb)) and inspects them whether they are plausible or not (i.e., classifies whether real or fake). The prediction that successfully fools the discriminator is selected as the final outcome.

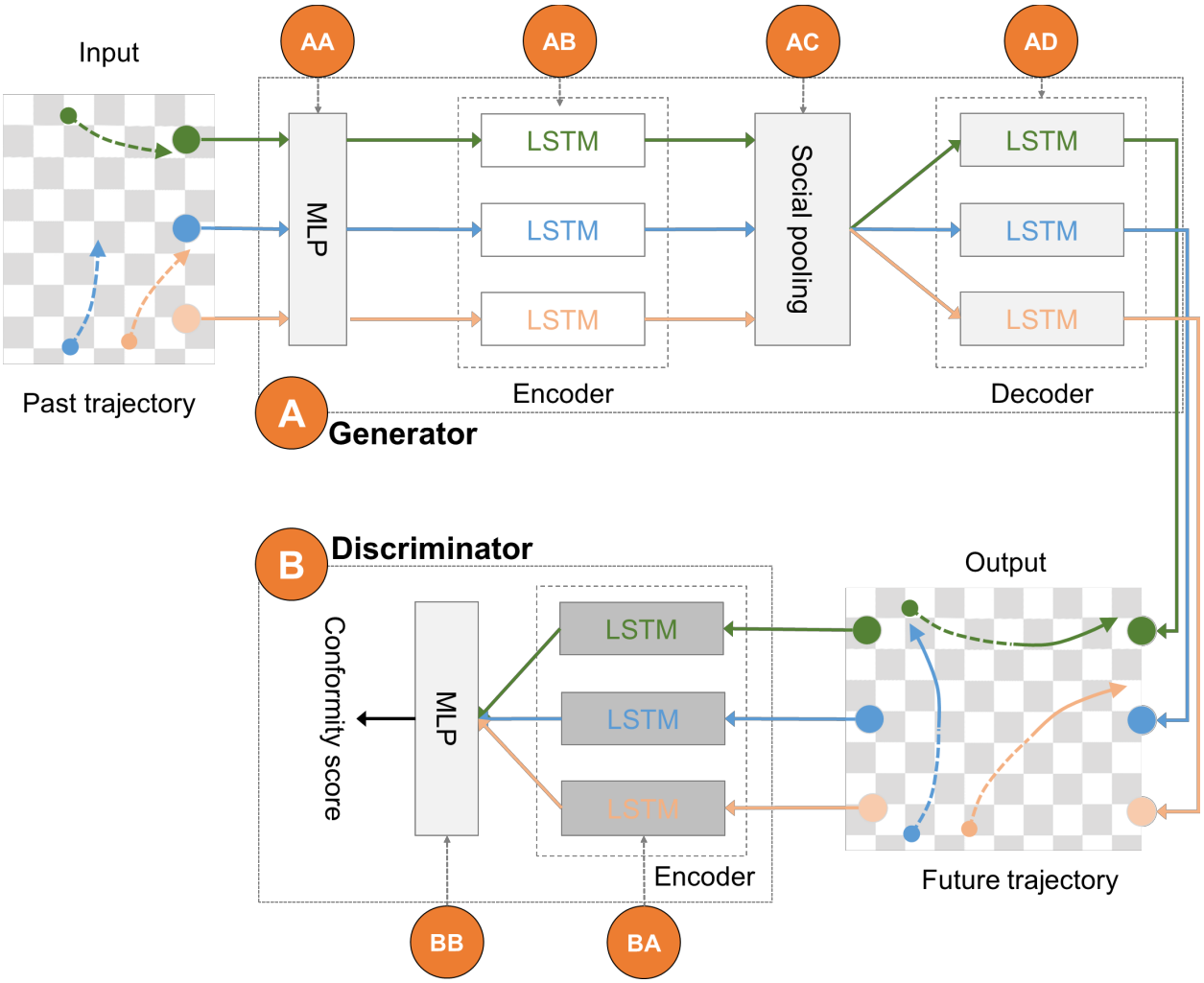

Figure 3.4 Network Architecture of S-GAN

3.2.2.2 Transfer learning of S-GAN

The authors developed a trajectory prediction model through transfer learning of the S-GAN. The following details were specifically considered: (i) parameter initialization, (ii) fine-tuning, and (iii) hyper-parameter tuning. This work started from the S-GAN model, which is pre-trained with the two benchmark datasets: (i) Eidgenossische Technische Hochschule Zurich (ETH, Pellegrini et al. 2010) and (ii) University of Cyprus (UCY, Leal-Taixe et al. 2014). As the most widely benchmarked datasets in trajectory prediction studies, the two datasets in total contain 1,536 human trajectories. They reflect various movement patterns such as crossing each other, collision avoidance, group forming, and dispersing (Alahi et al. 2016). Having such diverse data in pre-training was intended to prevent overfitting in the following fine-tuning process.

From that starting point (i.e., pre-learned weights), the fine-tuning with construction dataset was conducted to better fit the pre-trained model to construction settings. Specifically, I fine-tuned it with the integrated dataset (i.e., ETH + UCY + the construction dataset), rather than only with the construction dataset, so as to minimize the possibility of overfitting. In this tuning, the trajectories of construction mobile resources (e.g., worker, wheel loader, and excavator), annotated from 916 UAV-captured images, were used.

The farther prediction is achieved, the earlier safety intervention can be made. I thus modified the original prediction length (3.96 seconds=12 time-steps x 0.33 seconds) to 5.28 seconds (16 time-steps x 0.33 seconds) and particularly examined how observation-related hyper-parameters affects the model's final performance. Trajectory prediction is primarily based on the interpretation of targets' previous movement patterns. Thus, the properties of past trajectory must have a significant impact on the model's final performance. In this sense, this task additionally tuned the two observation-related hyper-parameters (i.e., observation length and sampling interval) with the following reasons.

- Observation length: a target's future trajectory is highly attributed to its previous movement pattern. The length of observation (i.e., how long observation the model will consume) must thus have a significant impact on a model's prediction performance. Thus, three different observation lengths were considered in this work: (i) 2.64 seconds (80 frames), (ii) 3.96 seconds (120 frames), and (iii) 5.28 seconds (160 frames).
- Sampling interval: the other hyper-parameter selected was sampling interval. This is because it controls the minuteness of input and output trajectories. With a denser sampling interval, the model can have finer input, but should take the burden of outputting denser prediction as well. On the other hand, with a sparser sampling interval, the model should have coarser input but can avoid such complexity. To examine which level of sampling interval would better fit for our problem, the authors considered four different sampling intervals: (i) 0.17 seconds (5 frames), (ii) 0.33 seconds (10 frames), (iii) 0.66 seconds (20 frames), and (iv) 1.33 seconds (40 frames).

3.2.2.3 Test result

For comparative evaluation of the twelve tuned models, the test on a construction dataset was followed. In this test, a total of 397 UAV-captured images was used and the trajectories of three object classes were considered: (i) worker, (ii) wheel loader, and (iii) excavator (Figure 3.5).

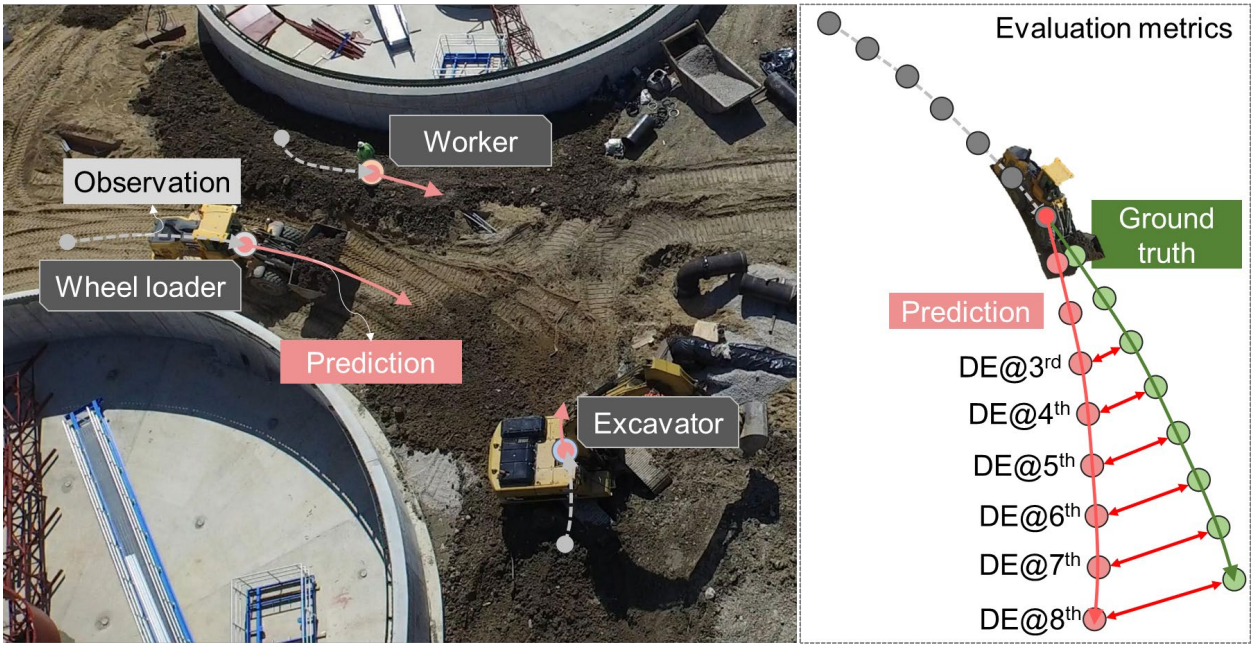

Figure 3.5 Trajectory Prediction Model's Test Dataset and Evaluation Metric

Note: DE stands for displacement error (unit=meters).

As evaluation metrics, average displacement error (ADE) and final displacement error (FDE), the typical two evaluation metrics to access trajectory prediction accuracy, were applied (Alahi et al. 2016; Gupta et al. 2018). The ADE is the average value of displacement errors (DEs, Euclidean distances) between ground truths and predictions over all predicted time-steps (i.e., average of DE@1^{st}~8^{th}, Figure 3.5) meanwhile the FDE is the distance between the predicted final destination and the ground truth destination at the end of the prediction period (i.e., DE@8^{th}, Figure 3.5). This test was intended to evaluate the pure performances of the tuned models, so I fed the models the ground truth of observation trajectories.

Table 1 summarizes the ADE and FDE results. Overall, the tuned models showed a promising prediction accuracy: all the ADEs were less than 0.9 meters and the FDEs were less than two meters. It was shown that the model of 0.66 seconds (20 frames) sampling interval and 3.96 seconds (120 frames) observation length has the highest accuracy in terms of both ADE and FDE: this model achieved the ADE of 0.45 meters and the FDE of 0.79 meters in this test. Considering this result, the authors adopted the model that showed the least error as the trajectory prediction module.

Table 3.1 ADE/FDE of Tuned Trajectory Prediction Models (Unit: Meters)

Sampling interval (unit: seconds)	Observation length (unit: seconds)		
	2.64	3.96	5.28
0.17	0.85/1.70	0.76/1.63	0.87/1.93
0.33	0.88/1.83	0.45/0.88	0.55/1.14
0.66	0.67/1.38	**0.45/0.79**	0.45/0.81
1.33	0.80/1.59	0.68/1.07	0.56/0.89

Note: left/right values are ADE/FDE, respectively; ADE/FDE in this table are average values of worker, wheel loader, and excavator; prediction lengths of all the models are 5.28 seconds.

3.3 Field Test

A field test was conducted to demonstrate the validity of the overall framework. It would have been ideal to test the proposed framework with mobile construction robots, since the robots are hardly available to date, this test employed a truck which is similar looking to an autonomous truck. Figure 3.6 illustrates the test environments and settings. In this test, the authors simulated the three types of movement patterns between a worker and a truck: (i) moving forward side by side (movement pattern #1); (ii) crossing each other side by side (movement pattern #2); and (iii) crossing each other in curves (movement pattern #3), as shown in Figure 3.6. The worker and the truck set off at the same time at the designated origins and followed the ground lines at a constant velocity (1.5 meters/second) until arriving at the designated destinations. The movement patterns were simulated three times per each. During this test, the authors flew a camera-mounted UAV over the testbed and ran the developed framework to predict the proximity between the targets (i.e.,

the metric distance between the worker and the truck). Lastly, the accuracy of the proximity outputs was evaluated by comparing it to the corresponding ground truth proximity.

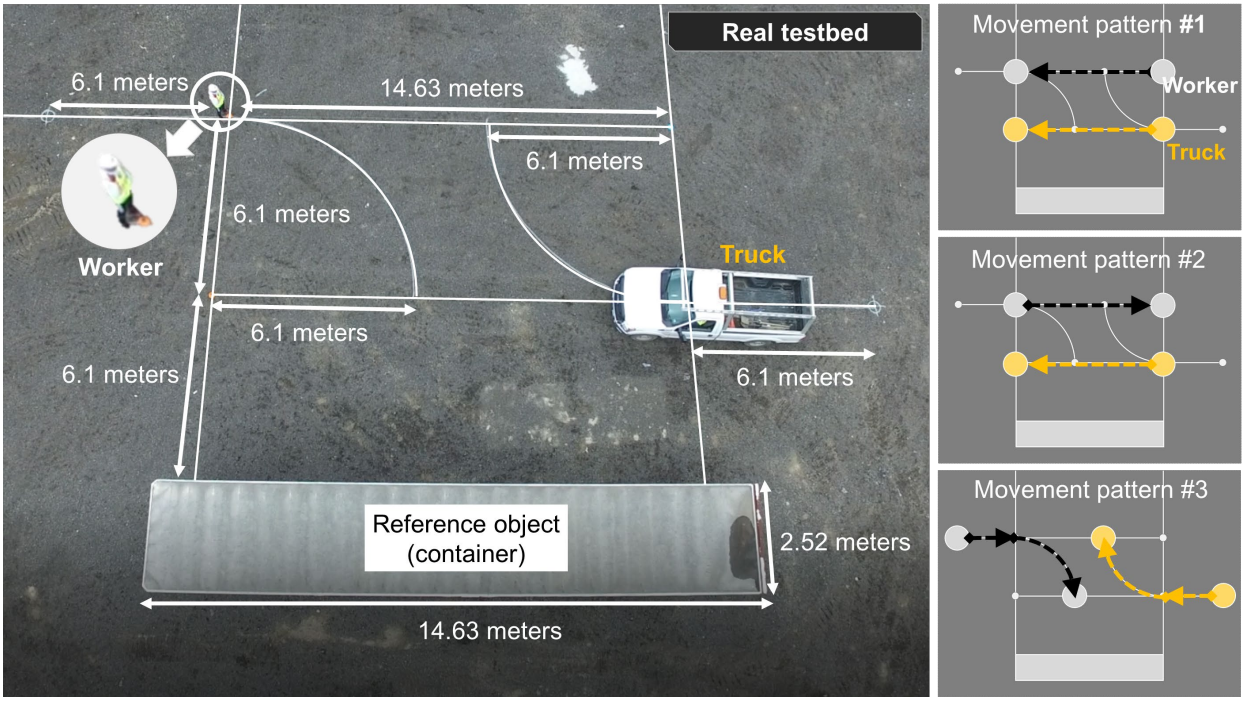

Figure 3.6 Field Test Settings

3.3.1 Measurement of ground truth proximity

To measure the ground truth proximity over all time-steps, I intentionally used ground lines and markers (Figure 3.6). The targets were ordered to follow a reference line at a constant velocity. Therefore, the origin-destination locations and times of a target were known so that the target's in-between locations and times could be measured by interpolation. In doing so, I measured all the ground truth locations of the targets over all time-steps and their ground truth proximity accordingly.

3.3.2 Evaluation metrics

To evaluate the accuracy of targets' predicted locations, the two displacement errors, Average Displacement Error (ADE) and Final Displacement Error (FDE), were applied. While the ADE and FDE represent the accuracy of predicted trajectory for each individual target, it does not directly represent the accuracy of predicted proximity between a pair of targets. Thus, in addition to the ADE and FDE, this test also evaluated Average Proximity Error (APE) and Final Proximity Error (FPE). The APE is the average value of the absolute differences between predicted proximity and ground truth proximity over all time-steps (Equation 3.1). Meanwhile, the FDE is the absolute difference between predicted proximity and ground truth proximity at the end of the prediction period (Equation 3.2). Lastly, this test also measured each module's operating time to evaluate its computational efficiency.

$$APE = \frac{1}{n} * \sum_{i=1}^{n} |P_g - P_p| \qquad \text{Equation 3.1}$$

Note: n=the number of cases; P_g=ground truth proximity; P_p=predicted proximity.

$$FPE = |P_{gf} - P_{pf}| \qquad \text{Equation 3.2}$$

Note: P_{gf}=ground truth proximity at the end of prediction period (i.e., after 5.28 seconds); P_{pf}=predicted proximity at the end of prediction period (i.e., after 5.28 seconds).

3.3.3 Proximity prediction result

In terms of ADE and FDE, the developed framework showed promising results. Overall, it achieved the ADEs for both the worker and the truck less than two meters, the FDEs less than 3.5 meters (Table 3.2). The ADE and FDE for the worker were 1.64 and 3.39 meters overall and those for the truck were 1.99 and 2.99 meters (Table 3.2). In line with the ADE and FDE results, the APE and FPE results were also promising. Overall, the framework achieved 0.95 meters APE and 1.71 meters FPE between the worker and the truck (Table 3.3). Also, the APEs between the worker and the truck for all three movement patterns were less than 1.5 meters, the FPEs less than 2.5 meters (Table 3.3).

Table 3.2 ADE and FDE for Truck and Worker (Unit: Meters)

Category	ADE		FDE	
	Worker	Truck	Worker	Truck
Movement pattern #1	1.76	1.84	3.06	2.32
Movement pattern #2	1.44	1.58	2.42	2.21
Movement pattern #3	1.73	2.54	4.68	4.45
Overall	**1.64**	**1.99**	**3.39**	**2.99**

Note: prediction length=5.28 seconds; ADEs and FDEs in this table are the average values for the three trials; overall values are the average for three movement patterns.

Table 3.3 APE and FPE between Truck and Worker (Unit: Meters)

Category	APE	FPE
Movement pattern #1	0.44	0.81
Movement pattern #2	1.23	1.94
Movement pattern #3	1.18	2.37
Overall	**0.95**	**1.71**

Note: prediction length=5.28 seconds; APEs and FPEs in this table are the average values for the three trials; overall values are the average for three movement patterns.

Notably, it was determined that to predict farther time-step is more challenging. Figure 3.7 illustrates the trend of proximity error (i.e., absolute difference between predicted proximity and ground truth proximity) as prediction time-step increases. As shown in Figure 3.7, for all movement patterns, the proximity errors continued to rise as the prediction time-step increases: on average, the framework showed the proximity error of 0.53 meters at 0.66 seconds prediction, but the error continued to climb as prediction time-step went farther, reaching to 1.71 meters (=the overall FPE, Table 3) at 5.28 seconds prediction (Figure 3.7).

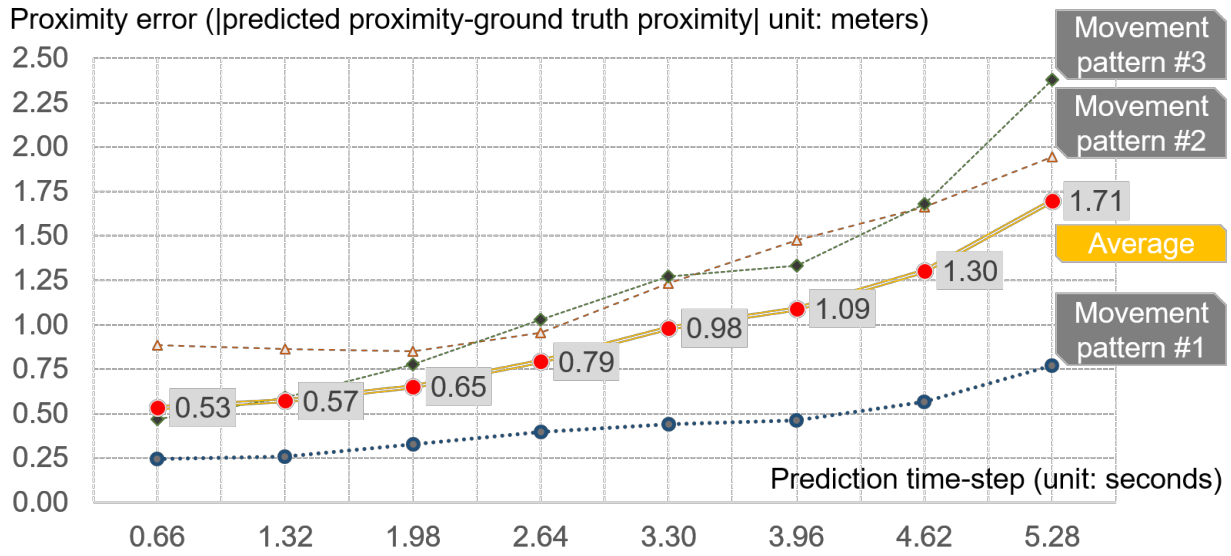

Figure 3.7 Trend of Proximity Error as Prediction Time-Step Increases

3.3.4 Operating time

Figure 3.8 illustrates the operating time of Modules 1 and 2. With a single graphic processing unit (GPU, NVIDIA Tesla K40), Module 1 (i.e., trajectory observation) spent 0.28 seconds per a frame (Figure 3.8(a)) and Module 2 (i.e., trajectory prediction) spent 0.12 seconds per a cycle (i.e., from taking a set of past trajectories to generating a set of future trajectories, Figure 3.8(b)). Given that this framework runs Module 1 at every 0.66 seconds (i.e., at 20 frames interval), it was able to perform trajectory observation with zero time-lag in computation. And overall, the framework demonstrated that it can update the future proximity for the next 5.28 seconds at every 0.66 seconds with 0.40 seconds time-lag in computation (i.e., 0.28 seconds for Module 1 + 0.12 seconds for Module 2, Figure 3.8(c)). It means that the framework can update future proximity for the next 4.88 seconds at every 0.66 seconds continuously (i.e., 5.28 seconds prediction – 0.40 seconds time-lag in computation).

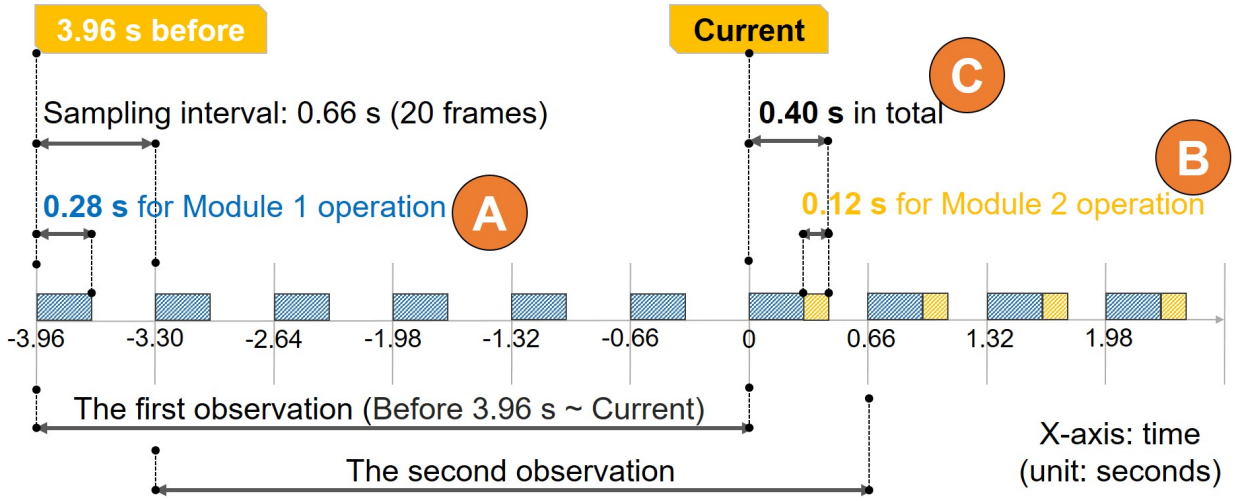

Figure 3.8 Operating Time of Modules 1 and 2

3.4 Discussions

As shown in the field test, the developed framework demonstrated a promising performance of proximity prediction in terms of both accuracy and speed. On the basis of the result, in this section, I present how this framework can better assist the collision avoidance between workers and mobile robots (or mobile equipment) at unstructured and dynamic construction sites. In addition, I discuss the implication of using GAN-based trajectory prediction DNN and lastly present potential improvement points for future studies.

3.4.1 Real-world applications to prevent contact-driven accidents by mobile objects

The framework showed that it can continuously update future proximity for the next 5.28 seconds at every 0.66 seconds within one-meter proximity error on average (computing time per update=0.40 seconds). This prediction performance can have a far-reaching significance beyond the detection of current proximity in accident prevention in that it enables pro-active safety interventions. For example, if a robot can be informed of whether a worker will be on the path or inside the action radius of itself in the future, the robot can make pro-active path planning and rerouting in advance. Likewise, it is also possible to provide an advance alert to workers via wearable devices (e.g., wrist band and smart safety glasses) so that the workers can take timely

evasive action. Assuming that an autonomous truck is approaching a worker at five meters per second, the framework can inform the worker and the autonomous truck of their potential collision 5.28 seconds before it happens. The worker then has around 25 meters of physical distance from the autonomous truck to easily avoid the collision without strain. These pro-active interventions would effectively reduce the chances of an impending collision between mobile robots and construction workers.

In addition, the developed framework also can be readily applied to other mobile objects such as motorized equipment and vehicle. This framework can detect mobile objects, such as excavator, wheel loader, and truck, and also, the scope of targets can be easily expanded through tuning of the object detection model with the additional training dataset. The framework can thus provide equipment operators and vehicle drivers with an alert in advance as well, helping to avoid a potential collision with workers and mobile robots.

In real-world applications, however, the quality and speed of network connection need to be further investigated and improved. The developed framework uses a camera-mounted UAV (or UAVs) to stream imagery input data to a computing device (e.g., a cloud server). Also, it needs wireless communication with robots and wearable devices to timely feedback. Therefore, in real-world applications, it is critical to ensure rapid data transmission from a computing device to a UAV (or UAVs), wearable devices, and robots. Leveraging 5G wireless network and internet of thing (IoT) cloud platform can be a promising solution to this end. The 5G wireless network would support real-time data transmission at data transfer rate of several gigabytes per second. Also, with the high-speed network connection, an IoT cloud platform could connect multiple UAVs, robots, and wearable devices to a cloud server, which would enable near real-time operation of proximity prediction as well as rapid communication with workers and robots.

To the fully automated operation of the proposed framework, the strategies for UAV operations need to be further studied. In the framework, UAV (or UAVs) plays a vital role in tracking target and reference objects. Therefore, future studies on how to capture mobile target objects and a stationary reference object simultaneously and continuously must be done. To this end, operating multiple UAVs and realizing real-time image stitching could be considered as a possible solution. Also, thorough field experiments need to be conducted in order to investigate how the elevation of UAV can impact proximity monitoring and prediction performance. The

higher elevation a UAV flies at, the wider the scene can be monitored. However, it can cause target objects to be seen too small, which can affect object detection performance and accordingly proximity monitoring and prediction accuracy.

3.4.2 Implication of using GAN-based DNN for trajectory prediction

GAN is basically an unsupervised generative model that makes plausible data from a noise input (e.g., Gaussian noise) based on probability distribution learned from real data (Goodfellow et al. 2014). The uniqueness of GAN that yields a highly competitive edge over other generative models (e.g., naïve Bayes, hidden Markov model, and Markov random fields) is the adversarial training between generator and discriminator. In GAN training, the generator tries to minimize min-max loss whereas the discriminator counteracts to maximize it (Equation 3.5). In this min-max game, both generator and discriminator get to improve while competing with each other. This adversarial training is known to better fit to understanding complex distributions of real data (e.g., images and speeches) than using a certain loss (objective) function manually devised.

$$\text{Min} - \text{Max Loss} = E_x[\log D(x)] + E_z[\log(1 - D(G(z)))] \qquad \text{Equation 3.5}$$

Note: D(x)=discriminator's estimate of the probability that real data instance x is real; E_x=expected value over all real data instances; G(z)=generator's output when given input z; D(G(z))=discriminator's estimate of the probability that a fake instance is real; E_z=expected value over all inputs to the generator.

The interesting fact is that the GAN can also be used for trajectory prediction which is basically a supervised learning problem. The S-GAN incorporates the GAN architecture and uses adversarial training so that it can enhance the capability to learn hidden distribution of mobile objects' diverse trajectories. More noticeably, the S-GAN leverages the GAN architecture in a conditional way such that it can still take prior information (i.e., past trajectory) as input and consume ground truth for network supervision. That is, instead of using noise input, it takes past trajectories and initializes the decoder with the prior information, thereby generating future trajectories more conformed to the past. Moreover, it uses L2 loss (Equation 3.6) in addition to the

min-max loss so that it can condition the decoder to generate the prediction closer to the ground truth. In these ways, the S-GAN could take advantage of both adversarial training and supervised learning, consequently resulting in a promising performance of trajectory prediction.

$$L_2 \text{ Loss} = \sum_{i=1}^{n} (Y_g - Y_p)^2 \qquad \text{Equation 3.6}$$

Note: n=dimension of output vector; Y_g=ground truth trajectory; Y_p=predicted trajectory.

However, the application of S-GAN presents several challenges, particularly in training. The adversarial training between generator and discriminator can be often stuck at local minima and in general takes a longer period than the training of normal DNNs. The single most important reason behind such challenges is the imbalance between generator and discriminator. For example, if the discriminator is too strong, then the generator training can easily fail due to vanishing gradients. On the other hand, if the generator easily defeats the discriminator, it tends to produce the most plausible output repeatedly, which can make the discriminator permanently trapped (called mode collapse).

Compared to dominant DNN architectures such as convolutional neural network (CNN) and recurrent neural network (RNN), GAN is a new kind of DNN. Certainly, there are still many chances to improve its trainability, which may include regularization using noise addition (Arjovsky and Bottou 2017), penalization of discriminator weights (Roth et al. 2017), and the use of advanced min-max loss (e.g., Wasserstein loss). The application of such advanced techniques would provide us with a better chance to leverage S-GAN (or other GAN-based DNNs) and to have a higher accuracy of proximity prediction thereby.

Another way to improve the prediction accuracy would include post-processing incorporating construction-specific knowledge. The S-GAN showed a promising accuracy of trajectory prediction in this study; however, it would not cover all the possible scenarios that can happen on construction sites and the prediction accuracy can deteriorate in those cases. The post-processing incorporating construction-specific knowledge, such as the average or maximum

velocity of each robot (or equipment), construction robots' pre-programmed collision avoidance behavior, and construction workers' collision avoidance behavior, can likely be used to refine predicted trajectory's velocity and direction, which could improve the overall accuracy of proximity prediction.

3.5 Conclusion

Following the previous study of proximity monitoring, I developed a DNN-based framework for proximity prediction leveraging trajectory prediction DNN (S-GAN). Also, I demonstrated the framework's validity in a field test: the framework achieved 0.95 meters average proximity error (APE) and 1.71 meters final proximity error (FPE) in predicting 5.28 seconds future proximity. During construction operations, contact-driven hazards by mobile robots (or mobile equipment and vehicles) can arise in various scenarios. For example, a navigating robot could suddenly change direction or an autonomous vehicle could reverse into a blind spot. In such unpredictable situations, proximity prediction would enable advance detection of impending collisions, thereby making pro-active interventions possible. Specifically, the predictive functionality would allow robots to make alternative path plans and reroute beforehand and would also enable providing advance alert to workers via wearable devices. These pro-active interventions would contribute to mitigating the chances of impending collisions between mobile robots (or mobile equipment and vehicles) and construction workers. Moreover, I apply GAN to trajectory prediction, which opens a new possibility of using GAN for construction applications.

CHAPTER 4

Semantic Relation Detection between Workers and Robots using a One-Stage Two-in-One DNN[3]

4.1 Introduction

The previous studies addressed the first agenda of the robotic hazard detection roadmap: monitoring proximity, which is an absolute precondition for a potential collision. Proximity between workers on foot and activated (mobile) robots is a must to detect potential collisions; however, concluding a hazard solely based on proximity could be inadequate. At times, field workers, robots, and equipment are meant to collaborate with one another at a close distance. Their proximity in such cases cannot be the sole determinant for a hazard, though it can be the precondition of one. Therefore, we need to consider associated entities' relations—whether they are co-working or not—to sensibly identify whether an event (e.g., a worker presents in action radius of an activated robot) is "cautious" or "hazardous" and thus needs an immediate intervention (Figure 4.1).

[3] This chapter is adopted from Kim, D., Goyal, A., Newell, A., Lee, S., Deng, J., and Kamat, V.R. (2019) "Semantic Relation Detection between Construction Entities to Support Safe Human-Robot Collaboration." *International Conference on Computing in Civil Engineering*, Atlanta, GA.

In the case of such circumstances, this study addressed the second agenda of the robotic hazard detection roadmap: semantic relation detection between workers and robots. To this end, I specifically investigated the potential of DNN-powered one-stage visual relation detection that can infer relations among construction entities directly from a site image (Figure 4.1). A DNN, when equipped with well-matched architecture, can connote both local and global features into a single composite feature map, which can potentially result in intuitive relation detection directly from an image—like a human-vision system (Newell and Deng 2017). Furthermore, a DNN, when trained with an enough data, can extract universal features that can lead to scalable relation detection to diverse construction environments (Girshick et al. 2015; He et al. 2015; Girshick 2015; Ren et al. 2017; Fang et al. 2018; Kim et al. 2018; Kolar et al. 2018).

Figure 4.1 DNN-Powered One-Stage Relation Detection

Despite such potential worthy of investigation, DNN-powered relation detection has received little to no attention in construction academia. In this study, I developed and evaluated several DNN-powered relation detection models with varying levels of task difficulty thereby examining in detail their potential for visual relation detection.

The remainder of this chapter is organized as follows: Section 4.2 explains the strong need for relation detection in identifying contact-driven hazards in co-robotic construction. Section 4.3 clarifies the research objective, describes the methods in detail, and presents the results with their implications. In section 4.4, I provide comprehensive discussion on the potential of DNN-powered semantic relation detection as well as the technical contributions of this study. Finally, a conclusion is drawn in Section 4.5.

4.2 Need of Relation Detection and Previous Approaches

In this section, I clarify the need for relation detection along with proximity monitoring in order to sensibly identify contact-driven robotic hazards. In addition, I present previous approaches for relation detection and their knowledge gaps, which this study intended to address.

4.2.1 Practical issue of proximity-based hazard detection and need of relation detection

Proximity is an apparent contributor to contact-driven accidents. However, solely relying on proximity when identifying a hazard is inadequate in practice. In construction, the proximity between a worker and a mobile object can arise unwittingly but proximity also happens intentionally while they are collaborating with each other. Take, for example, two simple cases: (i) a bricklayer has just unwittingly entered the action radius of an autonomous excavator and (ii) a bricklayer is finishing mortal joints beside a Semi-Autonomous Masonry (SAM) robot that is piling up bricks. The first case, without a doubt, constitutes a "hazard" where immediate intervention is called for. In contrast, the second is not a "hazard," since the two entities are working in intentional proximity. Considering such differences, proximity alone cannot always discern when a "hazard" is present. Therefore, we further need to consider relations between entities—whether they are co-working or not at the least—along with their proximity in order to sensibly identify a contact-driven robotic hazard. Otherwise, solely proximity-based hazard detection would result in frequent and unnecessary nuisances and halts of operation. Therefore, the necessity of relation detection is substantial in co-robotic construction environments where workers and robots commonly work alongside to one another.

4.2.2 Previous approaches for relation detection between construction objects

Despite the importance of relation detection, it has received attention in construction academia from only a small minority of studies. Cai et al. (2019) worked on relation detection between a worker and a piece of equipment—in cases where they are co-working and not—by analyzing positional and attentional cues. This study used the entities' locations (i.e., central coordinates of their bounding boxes) to describe their positional state, their head poses (i.e., yaw, roll, and pitch angles), body orientations, and body poses (e.g., standing or bending) in relation to their attentional states (Figure 4.2). The attributes of every entity in these cases were then compared to one another to formulate a feature descriptor that represents positional and attentional cues that are shared by the worker and the equipment (e.g., relative distance, direction, head yaw direction, body orientation, etc.). This type of engineering showed promising results coupled with a Long Short-Term Memory (LSTM)-based binary classifier. The coupling recorded precision and recall higher than 90% in a test on two construction videos. To our knowledge, this study is the first and only attempt to directly classify relations between construction objects using visually capturable information. Above all, it was original for this study to explicitly engineer a feature descriptor for relation detection.

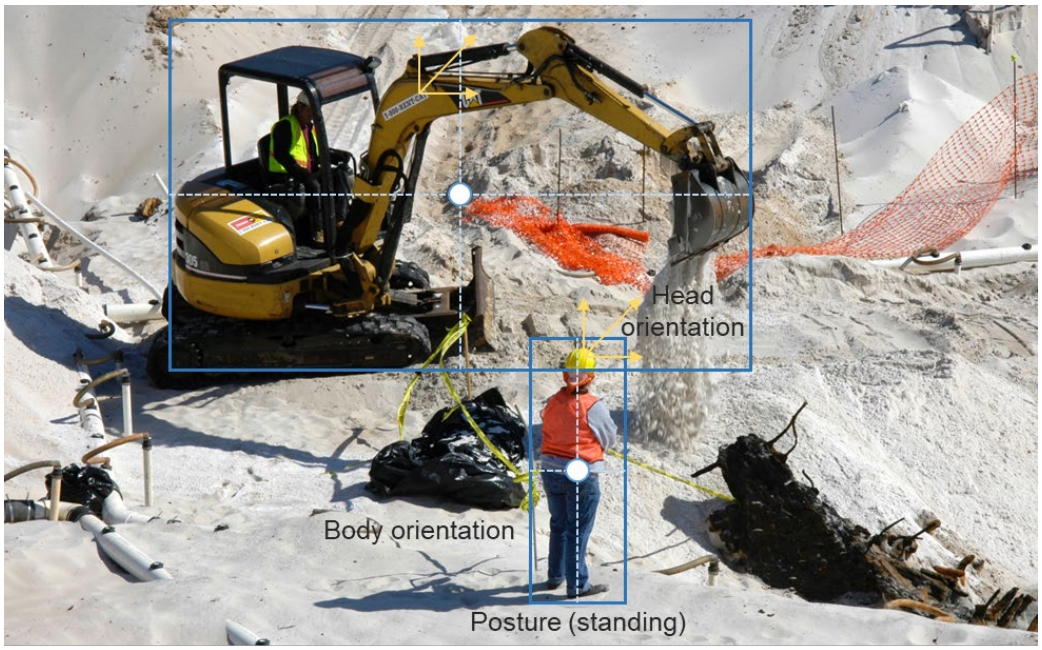

Figure 4.2 Examples of Hand-Engineered Features

It must not be overlooked that the hand-engineered approach described above is likely to involve scalability issues in real field applications. In other words, the approach could not show consistent accuracy under varied site conditions, neither could it reflect all possible scenarios (Brilakis et al. 2011; Park and Brilakis 2012; Memarzadeh et al. 2013; Kim et al. 2016). As many previous studies have pointed out, hand-engineering for feature extraction and description is often challenged by scene variances, which are highly common in construction environments. Hand-engineered methods could easily be misled under varying viewpoints and hand-engineered descriptors would not be capable of connoting all scene contexts potentially required for relation detection. Although the approach can be effective under certain conditions, it is not scalable under varied conditions, which highlights the need for feature extraction and description that can be universally applied to diverse construction environments.

Besides, we should not overlook the hand-engineered premise that all the pieces of information required for relation detection are available. Relation detection is a semantic inference, which requires comprehensive scene understanding. Accurate relation detection requires multiple pieces of information collected from each entity such as their location, pose, posture, movement direction, and speed. The previous study assumed such information as given. However, gathering these multiple pieces of information from an ongoing construction project is extremely challenging. Multimodal sensing and corresponding multiple pieces of data analytics would be required, which does not sound feasible in practice. Given these challenges, consideration is necessary for a method that more intuitively takes in the semantics of a scene—like human's vision system—rather than only inferring from fragmented pieces of information extracted from multimodal sensors.

4.3 DNN-Powered One-Stage Semantic Relation Detection

Toward more scalable and intuitive relation detection, I developed a DNN-powered one-stage relation detection model that can infer relation between a pair of construction objects directly from a site image. A DNN with deep Convolutional Neural Network (CNN) layers is capable of abstracting coarse-to-fine learned features of an input image. These learned features, when trained with a balanced number of data, can lead to more scalable relation detection in diverse construction environments that is more invariant to varied imaging conditions (Fang et al. 2018; Kim et al. 2018;

Kolar et al. 2018). Besides, a DNN with a flexible CNN architecture can connote both local and global features into a composite feature map at one-stage, which can potentially result in intuitive relation detection directly from an input image—like a human-vision system—and not rely on other sensing modalities and data analytics (Newell and Deng 2017). As such, it is worthwhile to investigate the potential of a DNN-based model as it can be an effective solution for scalable and intuitive relation detection.

To this end, I leveraged a unique DNN architecture [i.e., Pixel2Graph (Newell and Deng 2017)] specializing in multi-scale feature abstraction and one-stage relation detection. I started with a baseline model pre-trained with a benchmark dataset, Visual Genome (Krishna et al. 2017), and developed construction models that were fine-tuned with a balanced number of construction data with the architecture's complexity. Lastly, I tested my developed models with unseen construction datasets, thereby examining for scalability and potential in one-stage relation detection. Note that the validation for tuning hyper-parameters related to architecture, weight initialization, and the optimization algorithm was fully conducted in the original study (Newell and Deng 2017) and is thus not included in this study.

Herein, we developed three construction models that have varying levels of task difficulty, thereby examining the potential of DNN-powered one-stage relation detection in a phased manner:

- Model #1 (low level of difficulty)—Only Relation Detection (OnlyRel, Figure 4.3(a)): Object bounding boxes (bboxes) and classes are provided for all objects and the model only infers their relations.
- Model #2 (moderate level of difficulty)—Relation Detection + Object Classification (RelCls, Figure 4.3(b)): Object bboxes are provided for all objects,and the model classifies them and infers their relations.
- Model #3 (high level of difficulty)—Relation Detection + Object Detection (RelObj, Figure 4.3(c)): the model localizes and classifies all objects of interest and infers their relations.

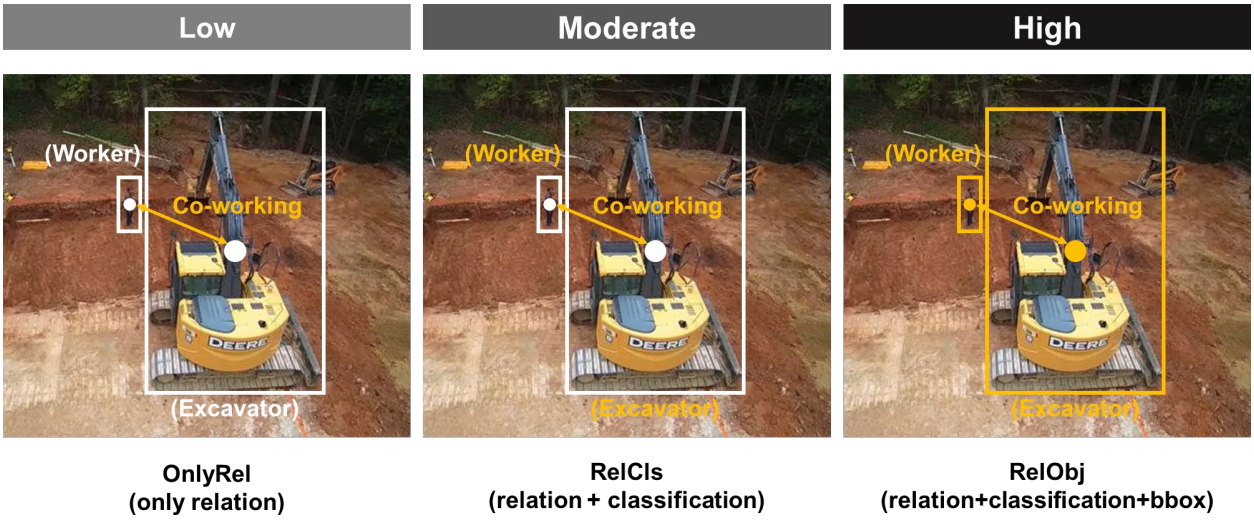

Figure 4.3 Different Levels of Task Difficulty: Low, Medium, and High

Note: White colored stands for given information; and orange colored stands for to be estimated.

The rest of this section provides the details of (i) the network's architecture, (ii) pre-training, fine-tuning, and test datasets, (iii) the model development process, (iv) the evaluation metrics, and (v) the training and test results with their implications.

4.3.1 Unique architecture of Pixel2Graph

Pixel2Graph (Newell and Deng 2017) has unique architecture consisting of three main modules: (i) a feature extractor (Figure 4.4(a)), (ii) a feature vector localizer (Figure 4.4(b)), and (iii) object and relation classifiers (Figure 4.4(c)).

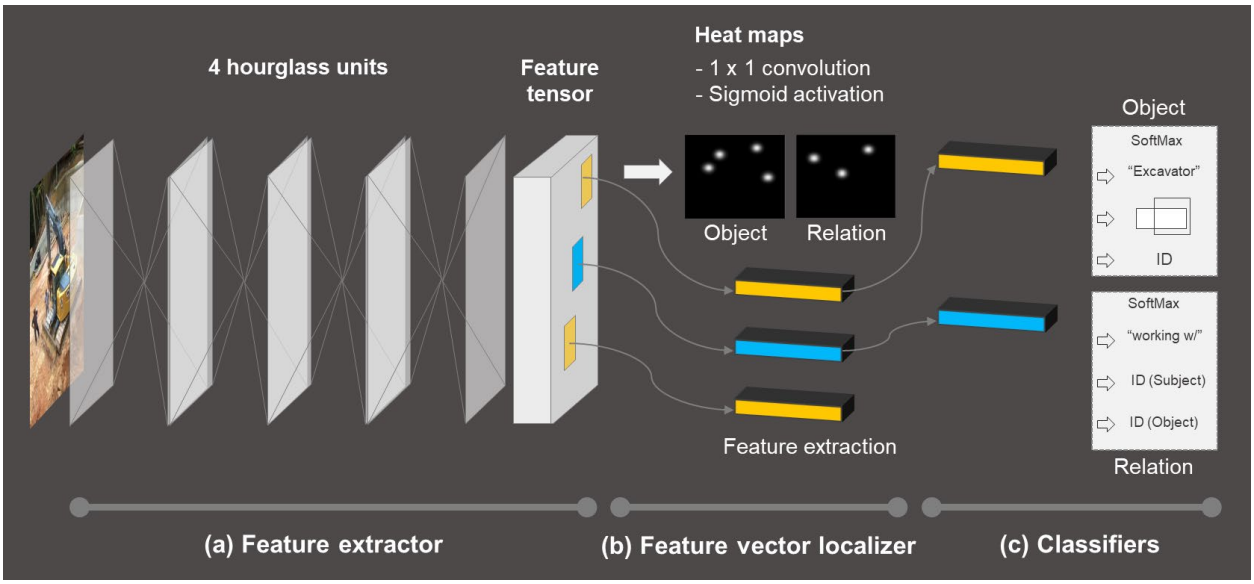

Figure 4.4 Network Architecture of Pixel2Graph

Compared to existing visual relation detection DNNs mostly supported by a region proposal network (RPN), this architecture has several distinctive features. In particular, the feature extractor, comprised of multiple hourglass units, enables the feature abstraction process to form both global and local features into a single feature tensor, which is more effective for understanding a scene as a whole (Newell and Deng 2017). In addition, associative embedding with likelihood heat maps for objects and relations allows for a one-stage, end-to-end process, which is capable of more cohesive and intuitive inferences about relations. The Pixel2Graph architecture is illustrated as follows and more details can be found in its original study (Newell and Deng 2017):

- Feature extractor (Figure 4.4(a)): The four hourglass network units stacked in a row take a whole image as input (width × height) and extract meaningful features from the unstructured input into a fixed-size 3D feature tensor (width × height × depth). An hourglass network unit is comprised of multiple convolutional layers in varying sizes with skip connections that enables encoding and decoding of feature extraction (Newell and Deng 2017). By repeating the cohesive abstraction process, the feature extractor can gather both global (e.g., connection between background and foreground objects) and local features (e.g., connection between foreground objects) into a single feature tensor,

which can be useful for relation detection as well as for object detection (Newell and Deng 2017).

- Feature vector localizer (Figure 4.4(b)): The feature vector localizer then specifies the potential locations of objects and their relations on the image's coordinates by analyzing the 3D feature tensor. The feature vector localizer generates likelihood heat-maps of objects and their relations independently through 1x1 convolution and sigmoid activation wherein each heat value represents the likelihood that an entity (i.e., object or relation) exists at the given location (Newell and Deng 2017). Based on the specified locations, the corresponding feature vectors of interest are selected and analyzed.

- Classifier (Figure 4.4(c)): The corresponding feature vectors are fed into the fully connected layer (details) and Soft-Max classifier (details), in which final classifications of (i) subject class (e.g., an excavator), relation (e.g., is working with), and (iii) object class (e.g., a worker) are performed.

4.3.2 Construction data collection and annotation

It is axiomatic in deep learning that the more diverse images a model trains with, the higher scalability and accuracy the model can achieve (Ren et al. 2017; Fang et al. 2018; Kolar et al. 2018). We thus collected a large volume of construction images and annotated them through a complete inspection. We collected videos from ongoing construction sites as well as from YouTube to cover a range of construction operations and backgrounds. Then, we sampled one image per each second from each video, thereby avoiding duplications in my dataset. In order to reduce the time and effort required for such a massive annotation, we leveraged web-based crowdsourcing with Amazon Mechanical Turk (AMT). We devised an annotation template that links the sampled images to the AMT server. This template lead AMT workers to annotate each object's bounding box, class label, and relations to others (Figure 4.5). We then followed these annotations with a complete inspection, thereby ensuring their validity. Figure 4.5 shows examples of several such annotated images. On each image, I labeled the bboxes and the classes of construction objects of interest and paired them by annotating relations between each pair of objects.

Table 4.1 summarizes the details of the prepared construction dataset. The total of 150 construction videos were collected each from a different site, from which 12,465 images were sampled at 0.1 sampling rate, and annotation followed. This dataset comprises seven classes of objects: (i) worker, (ii) excavator, (iii) truck, (iv) wheel loader, (v) roller, (vi) grader, (vii) scraper, and (viii) car. And among those objects, four classes of relation were identified and annotated: (i) guiding, (ii) adjusting, (iii) filling, and (iv) not working with. In total, 30,153 objects and 17,772 relations among them were annotated. Following the standards set by previous DNN studies (Redmon and Farhadi 2018), I considered 3,000 instances per each class sufficient for training. In other words, I was assured that this amount of construction data would be enough to fine-tune a baseline model and examine its potential.

Table 4.1 Details of Annotated Construction Dataset

Categories	Description
Total # of videos	150
Image sampling rate	0.1
Total # of annotated images	12,465
Total # of annotated objects	30,153
Total # of annotated labels	17,772
Ratio between co-working and not co-working labels	53:47

Note: among all the images, every 10^{th} images were sampled and annotated.

From here, I took measures to simplify the given problem. For the sensible identification of a contact-driven hazard, our focus in this study is to identify whether two associated objects are co-working or not—not to comprehend what the objects are doing. I thus reorganized the four relation classes into binary—(i) co-working and (ii) not co-working—by considering the first three classes (i.e., guiding, adjusting, and filling) as co-working (Figure 4.5). In this dataset, the even

ratio between co-working and not co-working (i.e., 53:47) was ensured, thereby avoiding biased training (Table 4.1).

Figure 4.5 Construction Dataset: Annotation Examples

4.3.3 Development of construction models

I developed three construction models with different levels of task difficulty—(i) OnlyRel, (ii) RelCls, and (iii) RelObj—via transfer learning from a baseline model developed in the original study (Newell and Deng 2017). I started from the baseline model pre-trained with a universal dataset—Visual Genome (Krishna et al. 2017)—that is the most widely used benchmark dataset for developing visual relation detection models. The Visual Genome (Krishna et al. 2017) contains around 108,077 frames including 3.8 million objects and 2.3 million relations. All the parameters of an empty Pixel2Graph architecture were initialized with pre-learned weights and continued to be updated through fine-tuning with the construction dataset. Of the entire dataset, 11,082 images (89%) were used for fine-tuning and the other 1,383 images (11%) were saved to be used for

testing. While splitting the construction dataset into these two categories, I ensured that there was no overlap in terms of site backgrounds or contexts, thereby avoiding potential overestimation in final testing.

4.3.4 Evaluation metric

I applied Recall@X as an evaluation metric, which is the one representative metric widely used in visual relation detection studies (Newell et al. 2017). Recall@X reports the fraction of ground truth tuples to appear in a set of top X estimations (Equation 4.1). Considering the diversity of the construction dataset, this study applied Recall@5.

$$\frac{1}{n}\sum_{i=1}^{n}\frac{CCinX}{X} \qquad \text{Equation 4.1}$$

Note: n stands for the total number of images; X=5 for Recall@5; CCinX stands for the number of correct classifications in top X estimations.

4.3.5 Fine-tuning and test results

The performances (i.e., Recall@5) of the three models (i.e., OnlyRel, RelCls, and RelObj) were promising, which are summarized in Table 4.2 and Figure 4.6.

Table 4.2 Recall@5s of OnlyRel, RelCls, and RelObj Models on Fine-Tuning and Test Datasets

Dataset	Recall@5s of the three models (unit=%)		
	OnlyRel	RelCls	RelObj
Fine-Tuning	90.89%	90.54%	92.96%
Test	90.63%	72.02%	66.28%

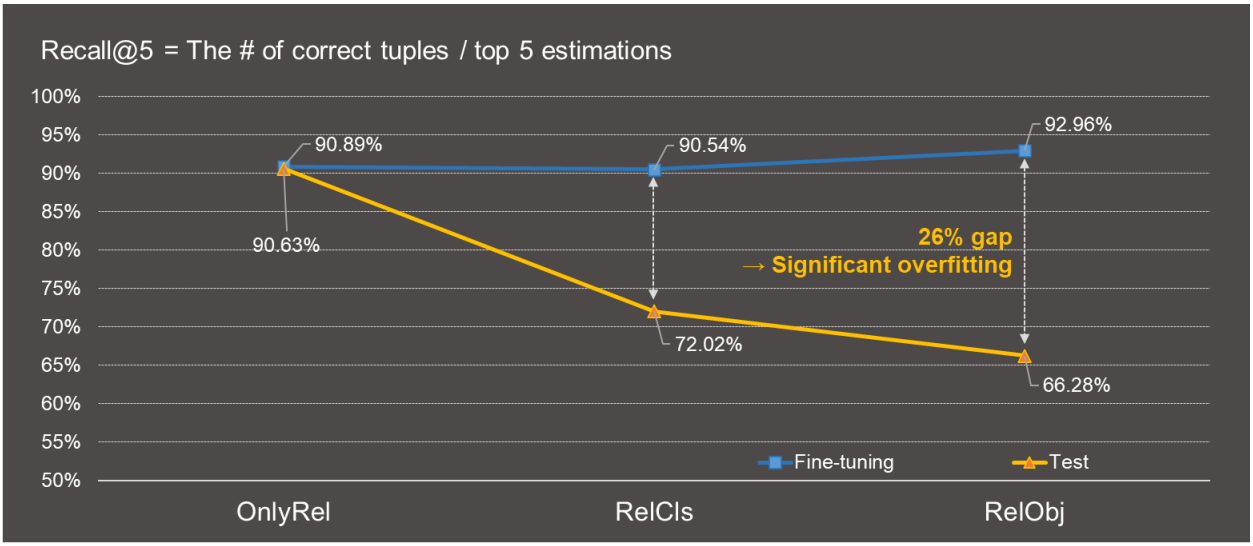

Figure 4.6 Recall@5s of OnlyRel, RelCls, and RelObj Models on Fine-Tuning and Test Datasets

- The OnlyRel model has low level of difficulty; it only infers the relation of each pair of entities (e.g., a worker and an autonomous robot), given the bboxes and classes of them. At this level, the fine-tuned model (OnlyRel) showed very promising results: it recorded 90.89% and 90.63% Recall@5s for fine-tuning and test datasets, respectively (Table 4.2 and Figure 4.6). As evidenced by the ignorable performance difference between the two datasets, there was no trace of overfitting and the model successfully scaled to the unseen test dataset with high performance same as on the fine-tuning dataset. From this result, it can be proven that a DNN, if equipped with well-fitted architecture (e.g., Pixel2Graph) and trained with an enough data, can have great potential for semantic relation detection. Given that the fine-tuned model can infer relations with high accuracy only from a single image, this result is noteworthy. However, it also needs to be noted that the OnlyRel model has a low level of task difficulty and is not technically one-stage as it needs object detection (i.e., bboxes and classes of targets) as input. Further investigations on the models having higher levels of difficulty were followed.

- Compared to the OnlyRel model, the RelCls model has a higher level of task difficulty; given bboxes of target entities, it infers their classes and relation at the same time. The RelCls model's Recall@5 on fine-tuning dataset was high as with the OnlyRel model:

it recorded 90.54% Recall@5 (Table 4.2 and Figure 4.6). On the other hand, RelCls recorded 72.02% Recall@5 on the test dataset, which was certainly lower than OnlyRel's (Table 4.2 and Figure 4.6). This result implies that relation detection accuracy can be hugely affected by the classification results of detected objects. The RelCls model abstracts object classification-related information as well as that for relation detection into one composite feature tensor. It turned out that such multiple information encoding in the current architecture is more challenging than focusing on one specific form of information (e.g., only relation-related). As a result, the object classification accuracy fell, in turn resulting in decreased relation detection performance.

- The RelObj model has the highest level of task difficulty; it performs bbox detection, object classification, and relation detection simultaneously in one single network. RelObj is a one-stage model. The RelObj model achieved 92.96% and 66.28% Recall@5s on training and validation datasets, respectively (Table 4.2 and Figure 4.6). As shown in Figure 4.7, the model's Recall@5 continued to improve with the fine-tuning dataset during the tuning session, converging at around 92.96%. This result showed that the network's architecture is capable of learning the situational context of a construction scene and has great potential for relation inference between construction objects at one-stage. On the test dataset, however, the model could not achieve as high performance as on the training dataset and ended up reaching a Recall@5 of 66.28%. Although the model showed steadily increasing performance with the test dataset during fine-tuning, it started to converge at the early stage. It was clear that learning the situational context along with object detection is more challenging. In particular, it turned out that successful training for one-stage relation detection requires a larger volume of fine-tuning data than for the OnlyRel or RelCls models. A significant discrepancy between the Recall@5s on fine-tuning and test datasets was confirmed—a typical symptom for overfitting. However, this result does not necessarily represent the potential maximum performance of the one-stage relation detection model. The 92.96% Recall@5 from the fine-tuning dataset clearly shows that the one-stage model has a high trainability but could not reach its maximum performance due to a limited number of fine-tuning data and resultant overfitting. A follow-up study with an augmented fine-tuning dataset would provide another chance to improve the RelCls model's

performance. Another consideration is to modify the original Pixel2Graph's architecture such that it can have multiple separated feature tensors for object detection and relation detection to improve its overall performance.

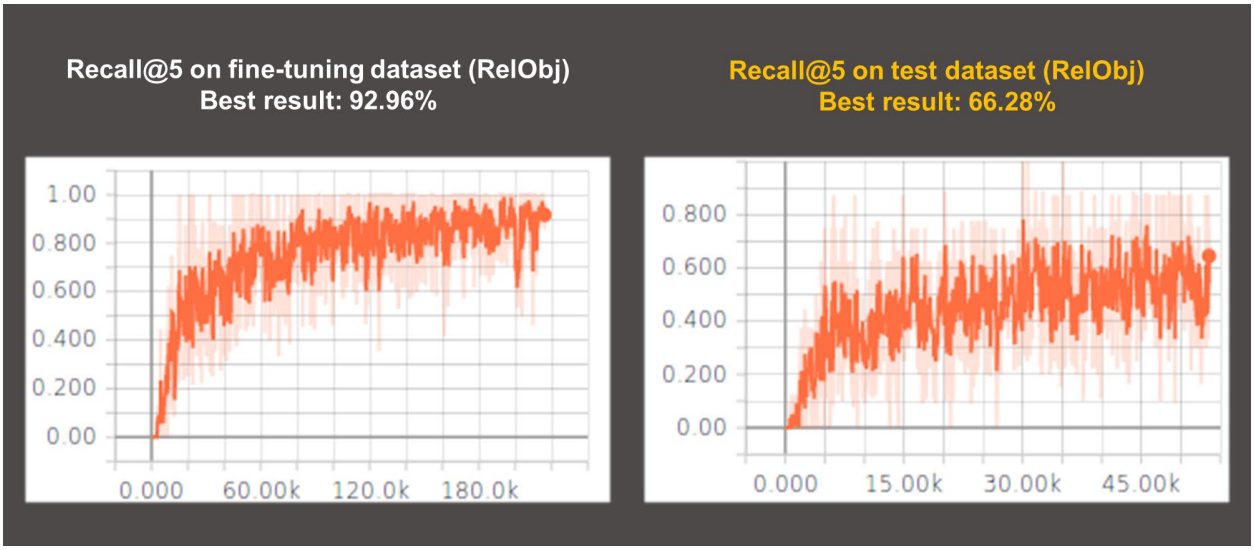

Figure 4.7 RelObj Model's Recall@5s for Relation Detection during Fine-Tuning

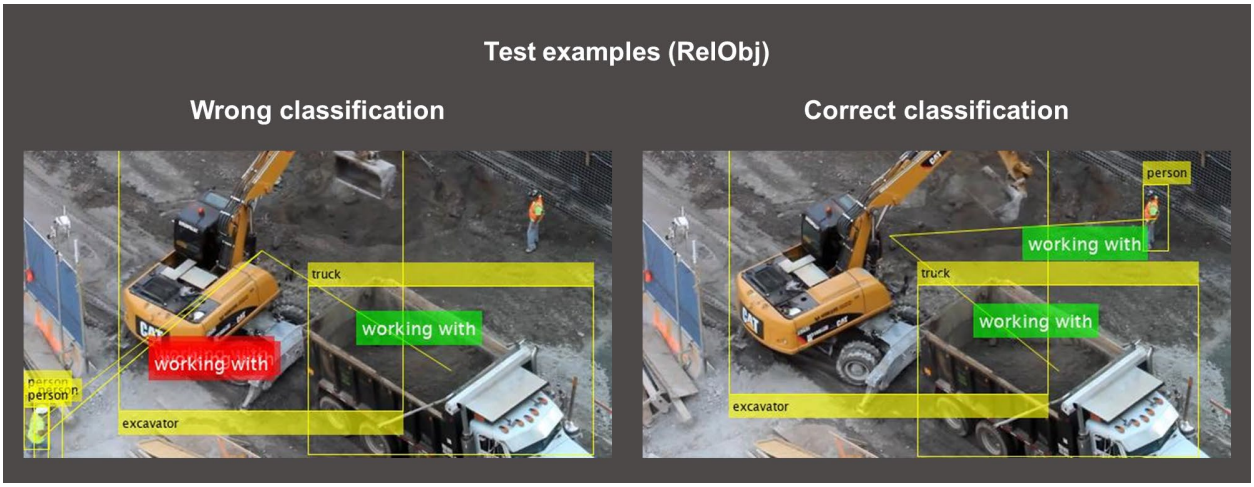

Figure 4.8 RelObj Model's Test Examples: Wrong and Correct Classifications

4.4 Discussions

Along with proximity monitoring (Chapters 2 and 3), relation detection (Chapter 4) is another essential element in robotic hazard detection. In co-robotic construction, it would be highly common and frequent that a worker is near activated (mobile) robots. However, concluding a hazard solely based on their proximity can be a hasty decision. At times, field workers and robots are meant to collaborate with one another and, if that is the case, their proximity cannot be the sole determinant for a hazard, even though it can be the precondition of one. Therefore, we must consider the associated entities' relation—whether they are co-working or not—to sensibly identify whether a situation is just "cautious" (e.g., a worker is near an activated robot and co-working with it) or "hazardous" (e.g., a worker is near an activated robot but not co-working with it) and therefore in need of immediate control.

However, relation detection—a semantic inference process—is not straightforward like object detection and requires holistic scene understanding. A possible way to achieve relation detection would be to figure out the multiple attributes of associated entities (e.g., location, proximity, pose, action, attention, etc.) and then infer their relation based on the attributes via a pre-defined logic. However, this would not be the best option in practice since it requires multiple sensing modalities and data analytics, which would be practically infeasible to implement in ongoing construction works. Besides, making scalable inference logic is challenging since a relation between two entities can be defined in countless ways.

Given the above, the results of DNN-powered visual relation detection models are noteworthy. From a single input image, the fine-tuned model (OnlyRel) could successfully encode relation-related information into one composite feature tensor and infer relations between target entities at 90.63% Recall@5. This result clearly shows the potential of DNN for visual relation detection, which is certainly more affordable and efficient than utilizing multimodal sensors. However, it was also noted that achieving a fully one-stage relation detection model (RelObj) is more challenging than a two-stage model (OnlyRel). Connoting all the pieces of information for both object detection and relation detection into one feature tensor was not very effective and a significant overfitting was confirmed during fine-tuning, unlike with the OnlyRel model. Considering that the one-stage model still has a high trainability (92.96% Recall@5 with the fine-tuning dataset), follow-up studies with architecture modification for separated tensor operation and

fine-tuning with an augmented dataset will help us examine the full potential of one-stage visual relation detection.

To my knowledge, this work is the first attempt to leverage a multi-in-one DNN architecture (i.e., object detection + relation detection) in the construction domain. Many visual site monitoring tasks (e.g., safety monitoring, progress monitoring, and/or quality control) may involve the need for multiple vision tasks such as object detection, relation detection, and semantic segmentation. Implementing these tasks by stages could involve cumulated errors and be computationally inefficient. In this sense, leveraging a one-stage solution could be worthwhile to investigate and this work could be a preceding example.

4.5 Conclusions

This study attempted to address the second agenda of the robotic hazard detection roadmap, which is relation detection between workers and activated (mobile) robots. Since it is highly common in co-robotic construction that activated (mobile) robots present near workers on foot, relation detection, along with proximity monitoring, is essential in order to sensibly classify whether a situation is "cautious" or "hazardous." To this end, I leveraged one-stage DNN architecture for visual relation detection and tested three models with different levels of task difficulty (i.e., OnlyRel, RelCls, and RelObj) in a phased manner.

The OnlyRel model (i.e., perform relation detection given bboxes and classes of associated entities) showed a promising result: it achieved 90.63% Recall@5 on unseen test dataset. However, development of a fully one-stage model, the RelObj model that performs object detection and relation detection simultaneously, proved to be more challenging. Further consideration of architecture modification is necessary so that the architecture can manage multiple feature tensors for object detection and relation detection independently in a single data flow. In addition, the strong need for additional training with an augmented fine-tuning dataset was confirmed. Follow-up studies with such measures will help us further examine the maximum potential of the one-stage visual relation detection DNN.

CHAPTER 5

3D Pose Estimation of Co-Workers using a Synthetic Construction Data-Trained 2D-to-3D Pose Transfer DNN

5.1 Introduction

This chapter introduces a study to address the third agenda of the robotic hazard detection roadmap—the 3D pose estimation of a co-worker (i.e., 3D reconstruction of human skeleton from a video). This study gets deeper into the case where a worker and a robot are collaborating at close proximity (e.g., a robotic arm piles up concrete masonry unit blocks while its co-worker is finishing mortar joints). In this case as well, the risk of a forcible collision exists, particularly from a robot's articulated body parts such as an arm. Herein, it must be ensured that any parts of the robot do not strike any parts of the co-worker. Even if a strike is unavoidable or has already happened, the robot's potential contact force must not exceed the worker's maximum allowable force. A collaborative robot must thus be able to sense and localize its co-worker's whole body precisely, thereby controlling its movement and limiting its potential contact force accordingly. To this end, 3D pose estimation of a co-worker from a co-bot's viewpoint is necessary. Therefore, this study aimed to develop a 3D pose estimation DNN with a custom-made synthetic construction dataset.

With diversified DNN architectures and enhanced computing power, the studies for DNN-powered visual 3D pose estimation have made large strides and introduced a variety of potential

approaches to date, including (i) direct estimation of a 3D pose from an RGB image (Li and Chan 2014; Tekin et al. 2017; Pavlakos et al. 2017); (ii) frame-wise 2D-to-3D pose transfer (Martinez et al. 2017; Chen and Ramanan 2017; Hossain and Little 2018); and (iii) video-wise 2D-to-3D pose transfer (Pavllo et al. 2019). Notably, because of its proven superiority over previous approaches, much research is now focused on the video-wise 2D-to-3D pose transfer (Pavllo et al. 2019). However, advanced DNN architectures are not straightforward to leverage for construction application due to a lack of training datasets. There are few public benchmark datasets, such as Human 3.6M (Ionescu et al. 2014) and HumanEva (Sigal et al. 2010), but they are limited in terms of diversity of pose and camera viewpoints. Moreover, these datasets are presumed free-to-use only for research purposes and would thus not be free for practical or commercial uses. Certainly, solely relying on public benchmark datasets would not be the best option for construction applications.

On the other hand, developing a construction training dataset on one's own is challenging. Labeling a 3D human pose from an image or video is not possible to do by hand; it requires the use of additional sensors, such as Inertial Measurement Unit (IMU) and/or a marker-based motion capture system (e.g., Optotrak^{TM}), along with video recording. However, applying such sensors to construction workers at an ongoing jobsite is cumbersome; IMU sensors can be easily degraded due to jamming by metallic objects and marker-based motion capture systems can be easily blinded if their line-of-sight is occluded at unstructured construction sites.

This study addressed the above issue by leveraging a custom-made synthetic construction dataset. In this study, I created synthetic construction videos—that come with automatically labeled 2D and 3D poses of virtual construction workers—by superimposing virtual construction workers onto diverse construction background images with varying lighting conditions, camera distances, and viewpoints. Leveraging the synthetic data, I trained, validated, and tested a state-of-the-art video-wise 2D-to-3D pose transfer DNN and confirmed comparable performance to the one trained with the Human 3.6M benchmark dataset.

The reminder of this chapter is organized as follows: Section 5.2 presents the major trends of the 3D pose estimation study where reasoning for the architecture selection is explained. Section 5.3 explains the detailed process of making synthetic construction data and the outcome. In section

5.4, training and test results of the synthetic data-trained construction model are presented with in-depth discussion. Finally, a conclusion is drawn in Section 5.5.

5.2 3D Pose Estimation DNN

Early work on visual 3D pose estimation with DNN adopted an end-to-end approach that estimates a human's 3D pose directly from an RGB image. With a variety of DNN architectures, this work (Li and Chan 2014; Tekin et al. 2017; Pavlakos et al. 2017) pioneered visual 3D pose estimation using a monocular camera. However, it was certain that estimating a 3D pose directly from a single image (or video) is far more challenging than estimating a 2D pose (Pavllo et al. 2019). Visual 3D pose estimation intrinsically involves (i) 2D pose estimation (i.e., localization of 2D joint locations on an image coordinate system) and (ii) 2D-to-3D inference (i.e., reconstruction of a 3D skeleton on a 3D global coordinate system from detected 2D joints). Processing both tasks without intermediate supervision—no matter what architecture is applied—has proven hard for a single DNN to learn (Pavllo et al. 2019). Visual 2D pose estimation has already been used in a number of commercial applications [e.g., ergonomic risk assessment (Kinetica Labs 2021)]. Granted such high accuracy of 2D pose estimation, it has become a trend to sperate visual 3D pose estimation into more manageable steps: (i) 2D pose estimation followed by (ii) 2D-to-3D inference.

Against this backdrop, a new family of visual 3D pose estimation has been attuned to 2D-to-3D pose transfer, which estimates 3D joint locations on global coordinates system from detected 2D joints locations (Pvallo et al. 2019). Prior works have proposed a range of potential solution to this: for example, Chen and Ramanan (2017) attempted to transfer detected 2D joint locations into 3D via a K-Nearest Neighbor (KNN) search in a pre-defined space where a large set of 2D and corresponding 3D joint locations were mapped; Pavlakos et al. (2017) used detected 2D joint locations with additional image features to get the 3D joint locations; alternatively, Zhou et al. (2016) estimated 3D joint locations from 2D by predicting each point's depth and Brau and Jiang (2016) estimated 3D joint locations by using priors about bone length and maximizing projection consistency with given 2D joint locations. These works' 3D pose estimation performances proved superior to the aforementioned end-to-end DNNs in visual 3D pose estimation competitions [e.g., 3D Poses in the Wild (3DPW) Challenge (3DPW 2021)].

Based on the above research, I opted to leverage a 2D-to-3D pose transfer DNN, VideoPose3D (Pavllo et al. 2019), which is illustrated in Figure 5.1 and Figure 5.2. VideoPose3D has two features that distinguish it from other 2D-to-3D pose transfer DNNs: (i) exploiting spatio-temporal information from video (Figure 5.1) and (ii) coordination-based efficient estimation (Figure 5.2). With these features, VideoPose3D performed visual 3D pose estimation better than state-of-the-art 2D-to-3D pose transfer DNNs developed in other studies (Zhou et al. 2016; Brau and Jiang; Chen and Ramanan 2017; Pavlakos et al. 2017).

- Exploiting spatio-temporal information from video (Figure 5.1): VideoPose3D adopts temporal convolution that takes a certain length of 2D joint sequences as input and reconstructs 3D skeletons of the same lengths. Most of the aforementioned 2D-to-3D pose transfer DNNs operate frame-wise; that is, they isolate a frame from the previous and next ones and estimate 3D joint locations at each frame, solely relying on the single frame information (i.e., 2D joint locations and other image features from a single time-step). However, a human's pose is continuous, not discrete. Leveraging temporal information (e.g., connectivity among previous, current, and next joint locations) along with spatial information (e.g., connectivity among different joint locations such as wrist, elbow, and shoulder) is certainly a more compatible choice.

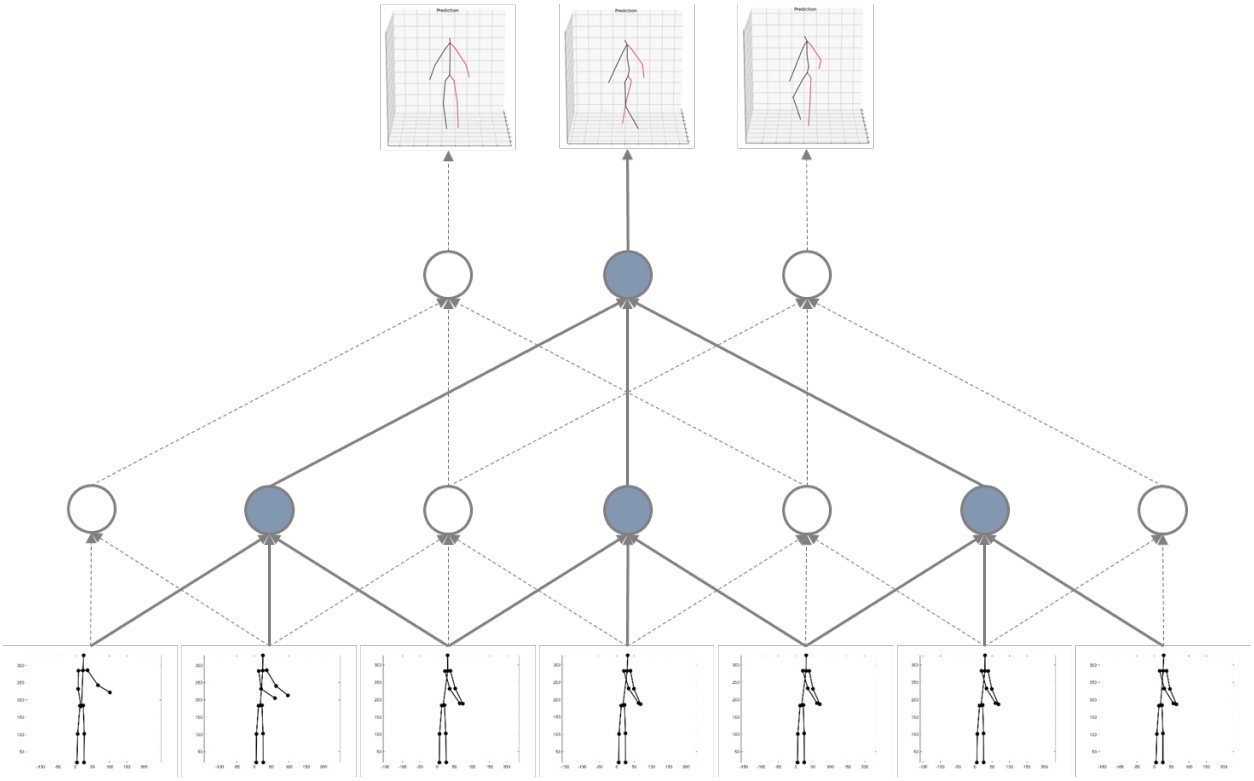

Figure 5.1 Dilated Temporal Convolution Concept

- Coordination-based efficient estimation (Figure 5.2): VideoPose3D directly describes a 3D human pose with 3D global coordinates whereas most of other DNNs apply a medium such as heatmaps for joint locations [Note: a separate heatmap is needed for each individual joint location (e.g., 17 joint locations=17 heatmaps)]. This capability allows for efficient 1D convolution over a coordinate time series which is more deterministic and, in particular, more efficient than applying 2D (or 3D) convolutions over multiple heatmaps. VideoPose3D thus shows potential for having higher accuracy with fewer parameters, allowing for faster training and inference. In addition, VideoPose3D applies batch normalization, dropout, and skip connections, thereby mitigating overfitting and faded gradient issues during training.

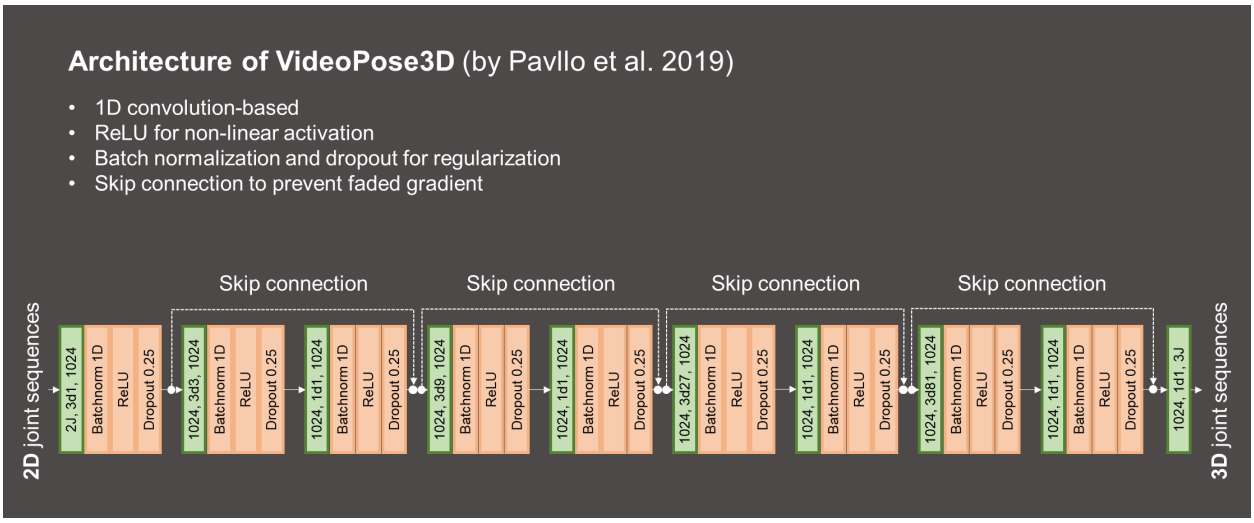

Figure 5.2 Network Architecture of VideoPose3D (Pavllo et al. 2019)

For these reasons, I leveraged VideoPose3D in this research and developed a construction model by training it with a custom-made synthetic construction dataset.

5.3 Synthetic Construction Data Generation

To generate a wide spectrum of synthetic construction data, I adopted a synthetic data generation framework: Synthetic Human for Real Tasks (SURREAL, Gul et al. 2018). The overall pipeline of SURREAL is illustrated in Figure 5.3.

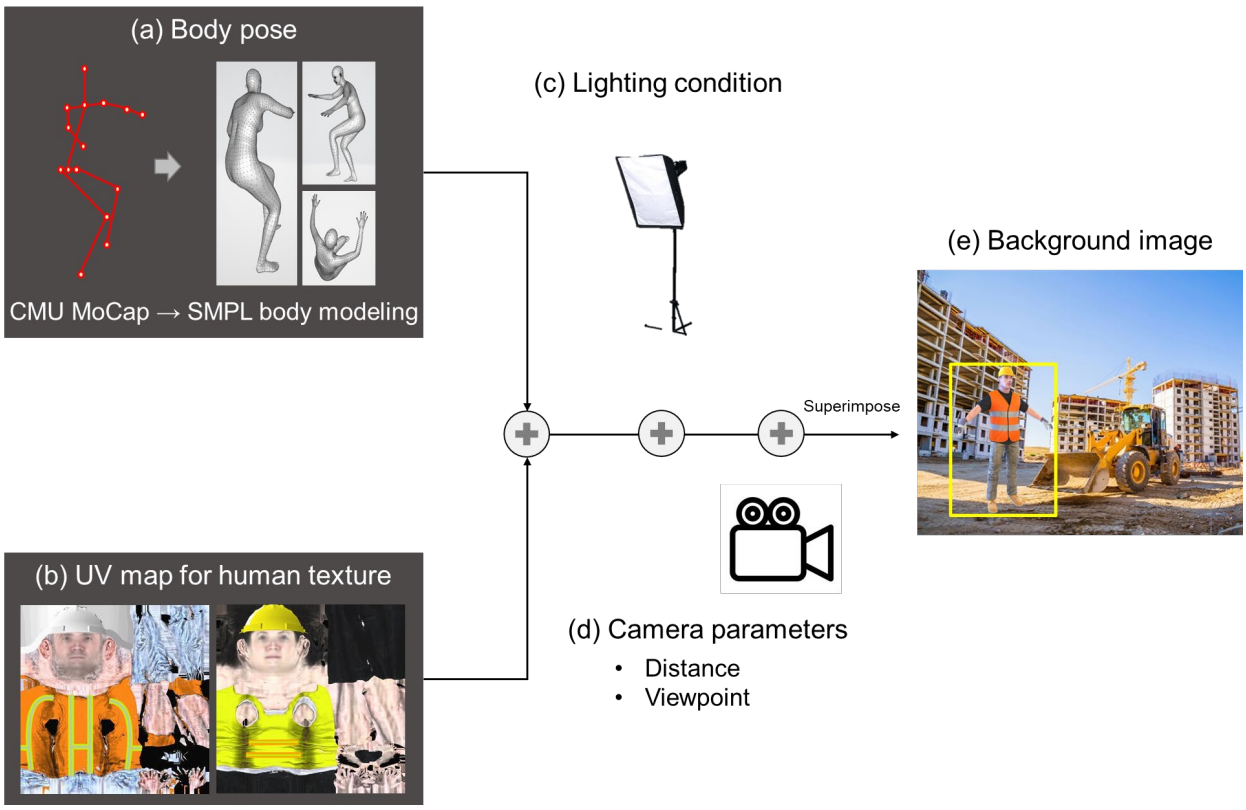

Figure 5.3 Overall Pipeline of SURREAL (Gul et al. 2018)

The SURREAL framework operates in Blender (an open-source animation package) with several components, including: (i) body pose, (ii) lighting condition, (iii) camera parameters, (iv) UV map for human texture, and (v) background image. This framework generates a synthetic video by superimposing a virtual 3D human model rendered from 3D sequences of motion capture data onto a background image from a random camera distance and viewpoint with a random lighting condition. Since we can freely modify the camera distance, viewpoint, and lighting conditions, we can generate a wide range of synthetic videos using a small set of motion capture data. Notably, since all components and related parameters involved in the video's generation are already given, the ground truths for 2D and 3D poses are automatically labeled while a video is rendered. I developed a synthetic construction dataset by using a public motion capture dataset [i.e., CMU MoCap dataset (CMU Graphics Lab 2021)], a UV map for construction worker clothing, and construction background images. All other parameters such as lighting conditions, camera parameters (i.e., distance and viewpoint), and body shape (e.g., thin or fat) were set to be randomly selected.

- Body pose (Figure 5.3(a)): As same with the original work (Gul et al. 2018), I used public motion capture data: the CMU MoCap dataset (CMU Graphics Lab 2021). CMU MoCap contains more than 2,000 sequences of 23 actions, totaling more than ten hours of 3D skeleton videos. The SURREAL framework takes a certain length of motion capture data as an input and, in turn, conducts 3D surface modeling using the Skinned Multi-Person Linear (SMPL) model (Loper et al. 2015). SMPL is a realistic articulated model of a human body learned from thousands of high-quality 3D scans; it represents a 3D human model with triangulated meshes (Loper et al. 2015).

- UV map for human texture (Figure 5.3(b)): To make a construction worker-looking virtual 3D human model, I created a 2D UV map for construction worker clothing, as shown in Figure 5.2(b). To do this, I added hardhat and safety vest with different color compositions (e.g., white, yellow, and orange) to a typical human UV map. The framework completes virtual 3D human modeling by putting the created clothing on the skinned 3D human model via UV texturing (i.e., 3D modeling process of projecting a 2D image to a 3D model's surface).

- Lighting condition (Figure 5.3(c)): In the SURREAL framework, a virtual 3D human model is illuminated using Spherical Harmonics with nine coefficients. In this research, I set the coefficient to be randomly selected from a uniform distribution between -0.7 and 0.7.

- Camera parameters (Figure 5.3(d)): In the first frame, the camera's location is determined such that the virtual 3D human model (center of hips) can be located at the center of the frame. I set the camera distance (i.e., the length between the camera lens and the virtual 3D human model) to be randomly selected from a normal distribution of eight meters mean and one meter standard deviation. In addition, the camera's yaw angle was set to be randomly selected.

- Background image (Figure 5.3(e)): For the background images, I collected 529 construction images from online sources. To use collected images as background, the presence of real workers was minimized during data collection. By superimposing the virtual 3D human model onto a construction background image at a randomly selected

lighting condition at randomly selected camara parameters, the SURREAL framework completes a synthetic construction video generation.

Leveraging the SURREAL framework with the CMU Mocap dataset, a UV map of construction worker clothing, and construction background images, I created a total of 529 synthetic construction videos each comprised of 243 frames and about eight minutes long. In total, 128,547 synthetic images were created. Examples of these videos are illustrated in Figure 5.4.

Figure 5.4 Examples of Created Synthetic Construction Videos

5.4 Training and Result

I trained the VideoPose3D DNN (Pavllo et al. 2019) architecture with the custom-made synthetic construction dataset. Of 529 synthetic construction videos, 352 were used in training and 177 in testing. I ensured there was no overlap of human poses between the training and test datasets, thereby preventing overestimation. For the VideoPose3D DNN training, I applied the same hyper-parameters as the original work, which is summarized in Table 5.1. For optimization, Adaptive Momentum Estimation (Adam) was applied with 0.1 initial momentum and 0.001 final momentum. The training was conducted for 80 epochs with 1,024 batch size at 0.001 learning rate. The model's input and output sequence lengths were both set to 243 frames.

Table 5.1 Training Hyper-Parameters for VideoPose3D DNN (Pavllo et al. 2019)

Categories	Description
Optimization algorithm	Adaptive momentum estimation (Adam)
Initial momentum	0.1
Final momentum	0.001
Epoch	80
Batch size	1,024
Learning rate	0.001
Learning rate decay	0.95
Input sequence length	243 frames
Output sequence length	243 frames

For evaluation, I applied the most widely-used evaluation metric for visual 3D pose estimation: Mean Per-Joint Position Error (MPJPE, unit=mm). MPJPE is the mean Euclidean distance between estimated joint locations and ground truth joint locations (Equation 5.1).

$$\frac{1}{n}\sum_{i=1}^{n}\sqrt{(x_e - x_{gt})^2 + (y_e - y_{gt})^2 + (z_e - z_{gt})^2} \qquad \text{Equation 5.1}$$

Note: n stands for the number of joints (in this study, 17 joint locations were used); x_e stands for estimated x-coordinate; x_{gt} stands for ground truth x-coordinate; y_e stands for estimated y-coordinate; y_{gt} stands for ground truth y-coordinate; z_e stands for estimated z-coordinate; z_{gt} stands for ground truth z-coordinate;

As shown in Figure 5.5, the synthetic data-based training was successful. The MPJPE for the training dataset smoothly decreased while training proceeding, which indicates that the custom-made synthetic data are fit to the VideoPose3D architecture. On the other hand, the model showed an unstable learning pattern for the test dataset at initial training stages. However, the MPJPE for the test dataset became stable soon after epoch #3 and continued to decrease during the training.

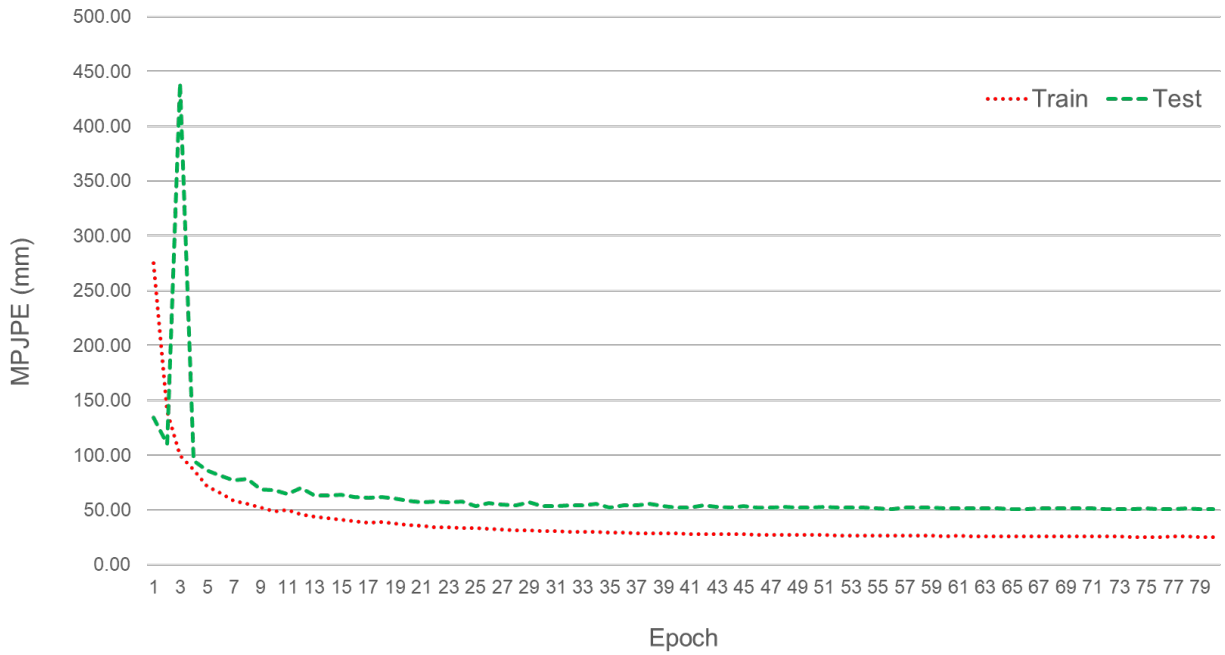

Figure 5.5 Training and Validation Logs

Table 5.2 summarizes the training and test results. The VideoPose3D model trained with only synthetic construction data showed promising results: it achieved 26.03 mm and 50.24 mm MPJPEs with the training and test datasets, respectively. There was a small difference in

performance between the training and test datasets and, as evidenced by the MPJPE patterns shown in Figure 5.5, no sign of overfitting was found. This data indicates that a balanced training is possible to achieve using only a synthetic dataset. Notably, the test performance of a synthetic data-trained model was highly comparable to that of a real data-trained model: the synthetic data-trained and Human 3.6M data-trained models' MPJPEs in their test dataset were 50.24 mm and 46.5 mm, respectively—only about a 4 mm difference. Since the two datasets take different human skeleton models [i.e., the way to represent a human skeleton is different even though they have the same number of joints (17)], this is not a perfectly fair comparison. Yet still, their closeness shows great potential of synthetic data for complementing existing public benchmark datasets. This result is noteworthy given the benefits of creating synthetic data: it is possible to create an unlimited number of images and labeling can be fully automated. To visually verify the performance of synthetic data-trained model, several test examples were illustrated in Figure 5.6.

Table 5.2 Training and Test Results: MPJPE (Unit: mm)

Dataset	MPJPE (unit=mm)
Train	26.03 mm
Test	50.24 mm

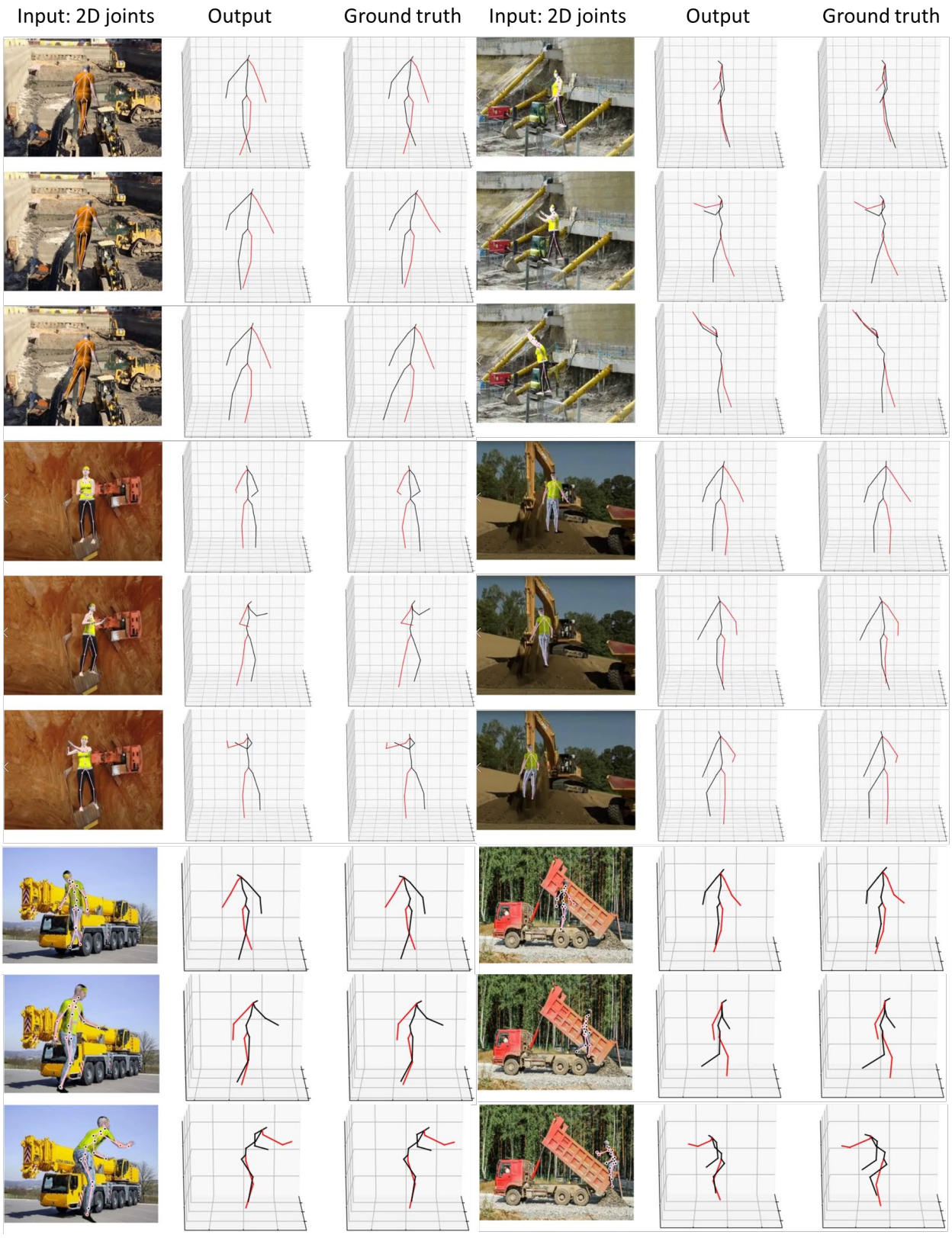

Figure 5.6 Test Examples

5.5 Conclusions

This study aimed to address workers' 3D pose estimation—the last component of the robotic hazard detection roadmap. Even in the case where a worker and a robot are collaborating, the risk of a forcible collision still exists, particularly from a robot's articulated body parts. It must be ensured that any parts of the robot do not strike any parts of the co-worker. Even if a strike is unavoidable or has already happened, the robot must be able to adjust its acceleration, thereby making its contact force not exceed the worker's maximum allowable force. To this end, accurate 3D pose estimation of co-workers is a must and I addressed it by developing a 3D pose estimation DNN with a custom-made synthetic construction dataset. In this study, the synthetic data-trained VideoPose3D model showed promising results (i.e., 50.24 mm MPJPE) which are comparable with that of a Human 3.6M data-trained model (46.5 mm MPJPE). This closeness indicates great potential for using synthetic data in 3D pose estimation DNN training. In follow-up studies, we would be able to examine how best to leverage both synthetic and real data together in 3D pose estimation DNN training (e.g., pre-training with a public benchmark dataset and then fine-tuning with a synthetic construction dataset), which would give us a lead on achieving optimal 3D pose estimation performance in real construction applications.

CHAPTER 6

Conclusions

6.1 Summary of Research

My Ph.D. research began with the overarching goal of developing and validating a visual site monitoring and hazard detection method that can complement robots' built-in safety functionalities. To achieve this end, I first established a three-phase robotic hazard detection roadmap and developed core technologies to implement it: (i) real-time proximity monitoring and prediction between workers on foot and activated mobile robots using camera-mounted UAVs; (ii) semantic relation detection between workers and robots using a one-stage two-in-one DNN; and (iii) 3D pose estimation of co-workers using a synthetic construction data-trained 2D-to-3D pose transfer DNN. To develop these technologies, I conducted four inter-related studies. A summary of these studies' results and their implications are as follows.

1. ***Real-Time Proximity Monitoring between Workers on Foot and Activated Mobile Robots using Camera-Mounted UAVs:*** This study addressed the first agenda of the robotic hazard detection roadmap: proximity monitoring between workers and activated mobile robots. To achieve less-occluded, real-time, and accurate proximity monitoring, I leveraged camera-mounted UAVs as imaging devices and developed a real-time visual

proximity monitoring method with DNN-powered computer vision and image processing techniques. In a real field test, the developed method can consistently monitor proximity between construction entities in a fully automated way at 0.61 m MADE (Mean Absolute Distance Error) and 4% MAPE (Mean Absolute Percentage Error) on average. This result demonstrates that the proposed method can serve as an effective proximity monitoring method in the conclusive hazard detection roadmap.

2. ***Proximity Prediction using a Conditional Generative Adversarial Network:*** Following-up the prior study for proximity monitoring, a prediction method for future proximity was developed using a conditional GAN [i.e., Social GAN (Gupta et al. 2019)]. In a field test, the developed method achieved 0.95 meters APE (Average Proximity Error) and 1.71 meters FPE (Final Proximity Error) in predicting 5.28 seconds future proximity. During construction operations, contact-driven accidents caused by mobile robots can happen anytime anywhere. Against such uncertainties, proximity prediction can have far-reaching effectiveness in accident prevention because it allows for more pro-active intervention—mitigating the chances of impending collisions between mobile robots (or mobile equipment and vehicles) and construction workers.
3. ***Semantic Relation Detection between Workers and Robots using a One-Stage Two-in-One DNN:*** This study addressed the second agenda of the robotic hazard detection roadmap: semantic relation detection between workers and activated mobile robots. Since it is highly common in co-robotic construction that activated (mobile) robots present near workers, relation detection along with proximity monitoring is necessary to sensibly classify whether a situation is "cautious" or "hazardous." To this end, I developed and tested one-stage two-in-one (object detection + relation detection) DNN models that had different levels of task difficulty (i.e., OnlyRel, RelCls, and RelObj). In the test on real field videos, the OnlyRel model (i.e., perform relation detection, given bboxes and classes of associated entities) showed promising performance (90.63% Recall@5); however, it was certain that the development of a fully one-stage model, the RelObj model that performs object detection and relation detection simultaneously, is more challenging (66.28% Recall@5). Further consideration on architecture modification and additional training with augmented fine-tuning dataset will help us

further examine the maximum potential of the one-stage visual relation detection DNN in construction applications.

4. ***3D Pose Estimation of Co-Workers using a Synthetic Construction Data-Trained 2D-to-3D Pose Transfer DNN:*** This study addressed the third agenda of the robotic hazard detection roadmap: 3D pose estimation of co-workers. The risk of forcible collision between workers and robots exists even in human-robot collaboration scenarios. To prevent potential accidents in collaborative work, 3D pose estimation of co-workers and the collaborative robot's self-adjustment of movement, acceleration, and contact force are a must. To this end, I developed a 3D pose estimation DNN with a custom-made synthetic construction dataset. In testing, the synthetic data-trained VideoPose3D model showed a promising result (i.e., 50.24 mm MPJPE), which is even comparable with a Human 3.6M data-trained model (46.5 mm MPJPE). These results indicate great potential for using synthetic data in 3D pose estimation DNN training. In follow-up studies, we would be able to examine the best way to leverage both synthetic and real data together in 3D pose estimation DNN training (e.g., pre-training with a public benchmark dataset and then fine-tuning with a synthetic construction dataset), which would give us a lead on achieving optimal 3D pose estimation performance in real construction applications.

6.2 Final Remark

I believe co-robotic construction is no longer the distant future. Robots will make their way into a variety of construction jobs and someday will become integral to construction. What co-robotic construction presents to the construction industry is promising and future construction work with robots will likely have a completely different horizon. Imagine a future blueprint of a construction site: a space where workers and robots harmoniously co-exist and work together. Most of the physically demanding, highly repetitive, and unpleasant tasks would be taken over by robots while human workers would focus on tasks that require fine dexterity and improvised decision making. It is certainly positive that robotic change will enable construction to be more productive and ergonomically safe. Additionally, as major roles of construction workers shift from bodily-dominant tasks to more intellectual ones, I believe the construction workforce can be more

attractive to prospective workers from a wide range of demographics and thus more inclusive. At de facto point of view, we must be capable of ensuring workers' safety first before embracing robotic solutions. I believe my Ph.D. research on visual site monitoring and hazard detection can partially contribute to making human-robot coexistence and collaboration in unstructured and dynamic construction sites safer and easier. Finally, the ensured safety and trust between robots and workers would contribute to promoting construction enterprises to embrace robotic solutions, boosting construction reformation toward innovative co-robotic construction.

6.3 Future Research Vision

My long-term research vision is to investigate what we need to prepare in advance to realize co-robotic construction and maximize its value. In order to settle co-robotic construction in the current labor-intensive construction industry, apart from safety issues, questions remain that warrant the attention of future research efforts.

In the rest of my research journey, I intend to find such questions and address them, including the following (Figure 6.1):

1. *What would be effective measures to achieve cohesive human-robot teaming in unstructured and dynamic construction environments? Would the existing master-salve type communication (e.g., speech recognition, hand gesture recognition, or hand guidance) be enough? Which type of interfacing technology will be required and what aspects (e.g., co-worker's psychophysiological responses) need to be considered in such new technologies?*
2. *How should construction work be re-designed (e.g., re-processing of labor-intensive work and work layout) and what would be ideal working conditions (e.g., robot's size and autonomy, leadership of collaboration, and the number of robots) to achieve optimized performance (e.g., productivity, safety, and quality)? Further, how can we design our work to improve workers' well-being and longevity in their careers?*

3. *How can we foster robot a collaboration-specialized workforce? How can we re-train existing workers and train new prospective workers for robot collaborative tasks? How can we lower the entry-barrier to robot collaborative tasks for aging construction workers and new prospective workers (e.g., young male/female workers)?*
4. *Can we apply existing industrial standards and regulations for the use of robots to unstructured and dynamic construction? To establish new standards and regulations for construction, which aspects need to be considered?*

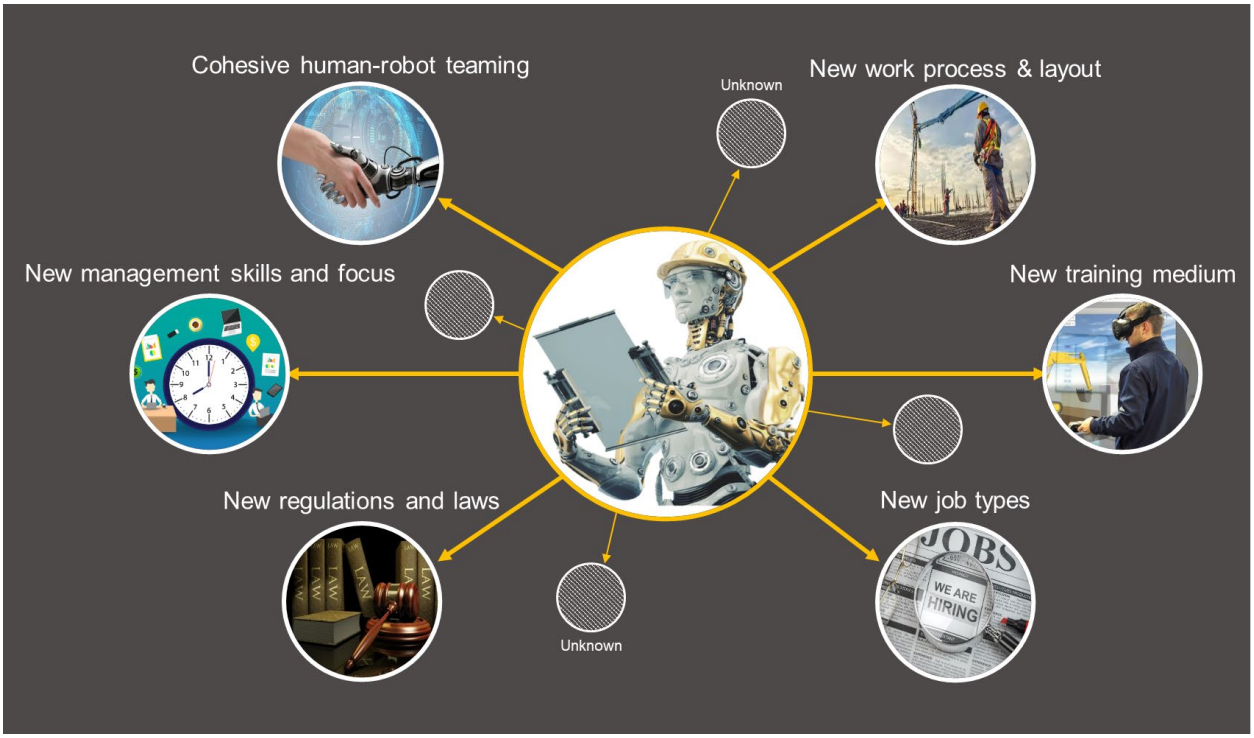

Figure 6.1 Long-Term Research Vision: Preparing The Big Wave of Co-Robotic Construction

BIBLIOGRAPHY

[1] Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., and Savarese, S. 2016. "Social lstm: Human trajectory prediction in crowded spaces." In Proc., 2016 IEEE Conference on Computer Vision and Pattern Recognition., 961-971. Las Vegas, NV: IEEE.

[2] Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., and Savarese, S. 2016. "Social lstm: Human trajectory prediction in crowded spaces." In Proc., 2016 IEEE Conference on Computer Vision and Pattern Recognition., 961-971. Las Vegas, NV: IEEE.

[3] Antonini, G., Bierlaire, M., and Weber, M. 2006. "Discrete choice models of pedestrian walking behavior." Transportation Research Part B: Methodological. 40 (8): 667-687. https://doi.org/10.1016/j.trb.2005.09.006.

[4] Antonini, G., Bierlaire, M., and Weber, M. 2006. "Discrete choice models of pedestrian walking behavior." Transportation Research Part B: Methodological. 40 (8): 667-687. https://doi.org/10.1016/j.trb.2005.09.006.

[5] Arjovsky, M., and Bottou, L. 2017. "Towards principled methods for training generative adversarial networks." In Proc., 5th Internaltional Conference on Learning Representations. Toulon, France. arXiv:1701.04862.

[6] Arjovsky, M., and Bottou, L. 2017. "Towards principled methods for training generative adversarial networks." In Proc., 5th Internaltional Conference on Learning Representations. Toulon, France. arXiv:1701.04862.

[7] Autodesk and Statista. 2021. "Building the future: Keeping up with a growing urban

population." <https://redshift.autodesk.com/building-the-future/> (Mar. 12, 2021)

[8] Autodesk. 2020. "100+ construction industry statistics." <https://constructionblog.autodesk.com/construction-industry-statistics/> (Mar. 12, 2021)

[9] Awolusi, I., Marks, E., and Hallowell, M. 2018. "Wearable technology for personalized construction safety monitoring and trending: Review of applicable devices." Automation in construction, 85 (Jan): 96-106. https://doi.org/10.1016/j.autcon.2017.10.010.

[10] Bock, T. 2015. "The future of construction automation: Technological disruption and the upcoming ubiquity of robotics." Automation in Construction. 59 (Nov): 113-121. https://doi.org/10.1016/j.autcon.2015.07.022.

[11] Brilakis, I., Park, M.W., Jog, G. (2011). "Automated vision tracking of project related entities." Advanced Engineering Informatics, 25, 713-724.

[12] Cai, J., Zhang, Y., and Cai, H. 2019. "Two-step long shor-term memory method for identifying construction acitivities through positional and attentional cues." Automation in Construction. 106(2019): 102886.

[13] Cardno, C. A. 2018. "Robotic Rebar-Tying System Uses Artificial Intelligence." Civil Engineering Magazine Archive. 88 (1): 38-39. https://doi.org/10.1061/ciegag.0001260.

[14] CAT Machine. (2012). "328D LCR Hydraulic Excavator." <http://s7d2.scene7.com/is/content/Caterpillar/C775795> (Aug. 22, 2017).

[15] Chen, C.H. and Ramanan, D. "3D human pose estimation = 2D pose estimation + matching." In Conference on Computer Vision and Pattern Recognition (CVPR) 2017.

[16] Chen, V. C., Li, F., Ho, S.-S., and Wechsler, H. 2006. "Micro-Doppler effect in radar: phenomenon, model, and simulation study." IEEE Transactions on Aerospace and electronic systems. 42 (1): 2-21. https://doi.org/10.1109/TAES.2006.1603402.

[17] Chi, S.H. and Caldas, C.H. (2012). "Image-based safety assessment: Automated spatial safety risk identification of earthmoving and surface mining activities." Journal of Construction Engineering and Management, 138(3), 341-351.

[18] CMU Graphics Lab. <http://mocap.cs.cmu.edu/> (Mar. 13, 2021).

[19] Commercial Construction Index (CCI). 2019. <https://mcsmag.com/skilled-labor-shortage-persistent-challenge/> (Mar. 12, 2021)

[20] CPWR, The Center for Construction Research and Training (2017). "Struck-by injuries and prevention in the construction industry." Silver Spring, MD, USA <www.cpwr.com> (Aug.

22, 2017).

[21] Cui, J., Liew, L. S., Sabaliauskaite, G., and Zhou, F. 2019. "A review on safety failures, security attacks, and available countermeasures for autonomous vehicles." Ad Hoc Networks, 90 (Jul): 101823. https://doi.org/10.1016/j.adhoc.2018.12.006.

[22] Devadass, P., Stumm, S., Brell-Cokcan, S., 2019. "Adaptive haptically informed assembly with mobile robots in unstructured environments", Proc. 36th Int. Symp. Autom. Robot. Constr. ISARC 2019. 469–476. https://doi.org/10.22260/isarc2019/0063.

[23] Dobson, R.J.., Brooks, C., Roussi, C., and Colling, T. (2013). "Developing an unpaved road assessment system for practical deployment with high-resolution optical data collection using a helicopter UAV." International Conference on Unmanned Aircraft Systems, Piscataway, NJ. USA.

[24] DuCarme, J. 2019. "Developing effective proximity detection systems for underground coal mines." Advances in Productive, Safe, and Responsible Coal Mining. 101-119. https://doi.org/10.1016/B978-0-08-101288-8.00003-1.

[25] Eschmann, C., Kuo, C.M., and Boller, C. (2012). "Unmanned aircraft systems for remote building inspection and monitoring." 6th European Workshop on Structural Health Monitoring, Dresden, Germany.

[26] Fang, Q., Li, H., Luo, X., Ding, L., Luo, H., Rose, T.M., and An, W. (2018). "Detecting non-harhat-use by a deep learning method from far-field surveillance videos." Automation in Construction, 85(2018), 1-9.

[27] Fernandez Galarreta, J, Kerle, N., and Gerke, M. (2015). "UAV-based urban structural damage assessment using object-based image analysis and semantic reasoning." Natural Hazards and Earth System Sciences, 15(6), 1087–1101.

[28] Gargoum, S. A., Karsten, L., El-Basyouny, K., and Koch, J. C. 2018. "Automated assessment of vertical clearance on highways scanned using mobile LiDAR technology." Automation in Construction. 95 (Nov): 260-274. https://doi.org/10.1016/j.autcon.2018.08.015.

[29] Girshick, R. (2015). "Fast R-CNN." International Conference on Computer Vision, Santiago, Chille.

[30] Girshick, R., Donahue, J., Darrell, T. (2015). "Region-based convolutional networks for accurate object detection and segmentation." IEEE Transactions on Pattern Analysis and Machine Intelligence, 38, 142-158.

[31] Global Infrastructure Outlook (GIO). 2020. "Forecasting infrastructure investment needs and gaps." <https://outlook.gihub.org/> (Mar. 12, 2021)

[32] Guiochet, J., Machin, M., and Waeselynck, H. 2017. "Safety-critical advanced robots: A survey." Robotics and Autonomous Systems. 94 (Aug): 43-52. https://doi.org/10.1016/j.robot.2017.04.004.

[33] Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., and Alahi, A. 2018. "Social gan: Socially acceptable trajectories with generative adversarial networks." In Proc., 2018 IEEE Conference on Computer Vision and Pattern Recognition., 2255-2264. Salt Lake City, UT: IEEE.

[34] Ham, Y.J., Han, K.K., Lin, J., and Golparvar-Fard, M. (2016). "Visual monitoring of civil infrastructure systems via camera-equipped unmanned aerial vehicles (UAVs): A review of related works." Springer, Visualization in Engineering, 4(1), 1-8.

[35] Han, K., Lin, J., and Golparvar-Fard, M. (2015). "A Formalism for utilization of autonomous vision-based systems and integrated project mdels for construction progress monitoring." Conference on Autonomous and Robotic Construction of Infrastructure. Ames, IA, USA.

[36] He, K., Zhang, X., Ren, S., and Sun, J. (2015). "Spatial pyramid pooling in deep convolutional networks for visual recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence, 37, 1904-1916.

[37] Helbing, D., and Molnar, P. 1995. "Social force model for pedestrian dynamics." Physical review E, 51 (5): 4282. https://doi.org/10.1103/PhysRevE.51.4282.

[38] Hossain, M.R.I. and Little, J.J. 2018. "Exploiting temporal information for 3D pose estimation." In European Conference on Computer Vision (ECCV) 2018.

[39] International Labor Organization (ILO), 2021. "World employment and social outlook." <https://www.ilo.org/global/about-the-ilo/multimedia/maps-and-charts/WCMS_337082/lang--en/index.htm> (Mar. 12, 2021)

[40] Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. 2014. "Human3.6m: Large scale datasets and predictive methods for 3D human sensing in natrual environment." Transaction on Pattern Analysis and Machine Intelligence (TPAMI) 2014.

[41] Jeelani, I., Asadi, K., Ramshankar, H., Han, K., and Albert, A. 2021. "Real-time vision-based worker localization & hazard detection for construction." Automation in Construction. 121(2021): 103448.

[42] Kerle, N., Fernandez Galarreta, J., and Gerke, M. (2014). “Urban structural damage assessment with oblique UAV imagery, object-based image analysis and semantic reasoning.” 35th Asian Conference on Remote Sensing. At Nay Pyi Taw, Myanmar. "

[43] Kim, D., Goyal, A., Newell, A., Lee, S., Deng, J., and Kamat, V. R. 2019a. "Semantic relation detection between construction entities to support safe human-robot collaboration in construction." 2019 ASCE International Conference on Computing in Civil Engineering., 265-272. Atlanta, GA: ASCE.

[44] Kim, D., Lee, S., and Kamat V. R. 2020. "Proximity prediction of mobile objects to prevent contact-driven accidents in co-robotic construction." Journal of Computing in Civil Engineering, 34(4).

[45] Kim, D., Liu, M., Lee, S., and Kamat, V. R. 2019b. "Trajectory prediction of mobile construction resources toward pro-active struck-by hazard detection." In Proc., International Symposium on Automation and Robotics in Construction., 982-988. Banff, AB, Canada.

[46] Kim, D., Liu, M., Lee, S., and Kamat, V. R. 2019c. "Remote proximity monitoring between mobile construction resources using camera-mounted UAVs." Automation in Construction. 99 (Mar): 168-182. https://doi.org/10.1016/j.autcon.2018.12.014.

[47] Kim, D., Yin, K., Liu, M., Lee, S.H., and Kamat, V.R. (2017). "Feasibility of a drone-based on-site proximity detection in an outdoor construction site." IWCCE 2017, Seattle, WA, USA.

[48] Kim, H.J., Bang, S.D., Jeong, H.Y., Ham, Y.J., and Kim, H.K. (2018). "Analyzing context and productivity of tunnel earthmoving process using imaging and simulation." Automation in Construction, 92(2018), 188-198.

[49] Kim, H.J., Kim, K.N., and Kim, H.K. (2016). "Vision-based object-centric safety assessment using fuzzy inference: Monitoring struck-by accidents with moving objects." Journal of Computing in Civil Engineering, 30: 04015075.

[50] Kim, K.N., Kim, H.J., and Kim, H.K. (2017). "Image-based construction hazard avoidance system using augmented reality in werable device." Automation in Construction, 83(2017), 390-403.

[51] Kim, P., Chen, J., and Cho, Y. K. 2018. "SLAM-driven robotic mapping and registration of 3D point clouds." Automation in Construction. 89 (May): 38-48. https://doi.org/10.1016/j.autcon.2018.01.009.

[52] Kinetica Labs. <https://kineticalabs.com/> (Mar. 13, 2021).

[53] Kolar, Z., Chen, H., and Luo, X. (2018). "Transfer learning and deep convolutional neural networks for safety guardrail detection in 2D images." Automation in Construction, 89(2018), 58-70.

[54] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., Bernstein, M.S., and Fei-Fei, L. 2017. "Visual genome: Connecting language and vision using crowdsourced dense image annotations." International Journal of Computer Vision, 123.1(2017), 32-73.

[55] Lattanzi, D., and Miller, G. 2017. "Review of robotic infrastructure inspection systems." Journal of Infrastructure Systems. 23 (3): 04017004. https://doi.org/10.1061/(ASCE)IS.1943-555X.0000353.

[56] Leal-Taixé, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B., and Savarese, S. 2014. "Learning an image-based motion context for multiple people tracking." In Proc., 2016 IEEE Conference on Computer Vision and Pattern Recognition., 3542-3549. Las Vegas, NV: IEEE.

[57] Li, J., Wang, Y., Zhang, K., Wang, Z., and Lu, J. 2019. "Design and analysis of demolition robot arm based on finite element method." Advances in Mechanical Engineering. 11 (6): 1687814019853964. https://doi.org/10.1177/1687814019853964.

[58] Li, S. and Chan, A.B. 2014. "3D human pose estimation from monocular images with deep convolutional neural network." In Asian Conference on Computer Vision (ACCV), Springer.

[59] Liang, C.J., Lundeen, K.M., McGee, W., Menassa, C.C., Lee, S., and Kamat, V.R. 2019. "A vision-based marker-less pose estimation system for articulated construction robots." Automation in Construction. 104(2019): 80-94.

[60] Lin, J., Han, K., Fukuchi, Y., Eda, M., and Golparvar-Fard, M. (2015). "Model based monitoring of work in progress via images taken by camera equipped UAV and BIM." International Conference on Civil and Building Engineering Informatics. Tokoy, Japan.

[61] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. 2014. "Microsoft coco: Common objects in context." In Proc., European conference on computer vision., 740-755. Zurich, Swiss: Springer

[62] Liu, J., and Li, G. 2018. "Research on the development of 3D printing construction industry

based on diamond model." Innovative Technology and Intelligent Construction., 164-176. Reston, VA: ASCE.

[63] Liu, M., Han, S. and Lee, S. 2017. “Potential of convolutional neural network-based 2D human pose estimation for on-site activity analysis of construction workers.” ASCE International Workshop on Computing in Civil Engineering 2017, Seattle, WA.

[64] Loop, C., and Zhang, Z. 1999. "Computing rectifying homographies for stereo vision." In Proc., 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition., 125-131. Fort Collins, CO: IEEE.

[65] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J. 2015. "SMPL: A skinned multi-person linear model." ACM Transactions on Graphics (TOG). 34.6: 1-16.

[66] Luo, H., Wang, M., Wong, P., Cheng, J.C.P. 2020. "Full body pose estimation of construction equipment using computer vision and deep learning techniques." Automation in Construction. 110(2020): 103016.

[67] Marks, E. and Teizer, J. (2012). "Proximity sensing and warning technology for heavy construction equipment operation" Construction Research Congress 2012, West Lafayette, IN, USA.

[68] Martinez, J., Hossain, R., Romero, J., Little, J.J. 2017. "A simple yet effective baseline for 3d human pose estimation." In International Conference on Computer Vision (ICCV) 2017

[69] Mckinsey Global Institute (MGI), 2017. "Reinventing construction: A route to higher productivity."

[70] Memarzadeh, M., Golparvar-Fard, M., and Niebles, J. C. 2013. "Automated 2D detection of construction equipment and workers from site video streams using histograms of oriented gradients and colors." Automation in Construction. 32 (Jul): 24-37. https://doi.org/10.1016/j.autcon.2012.12.002.

[71] Memarzadeh, M., Golparvar-Fard, M., Niebles, J.C. (2013). "Automated 2D detection of construction equipment and workers from site video streams using histogram of oriented gradients and colors." Automation in Construction, 32, 24-37.

[72] Michael, N., Shen, S., Mohta, K., Kumar, V., Nagatani, K., Okada, Y., Kiribayashi, S., Otake, K., Yoshida, K., Ohno, K., Takeuchi, E., and Tadokoro, S. (2014). “Collaborative mapping of an earthquake damaged building via ground and aerial robots.” Journal of Field and Service Robotics, 29(5), 832-841.

[73] Moon, S., Becerik-Gerber, B., and Soibelman, L. 2019. "Virtual Learning for Workers in Robot Deployed Construction Sites." Advances in Informatics and Computing in Civil and Construction Engineering., 889-895.

[74] Newell, A. and Deng, J. 2017. "Pixels to graphs by associative embedding." Advances in Neural Information Processing Systems, 2171-2180.

[75] Oskouie, P., Becerik-Gerber, B., and Soibelman, L. (2015). "A data quality-driven framework for asset condition assessment using LiDAR and image data." Journal of Computing in Civil Engineering, 2015, 240–248.

[76] Park, J., Marks, E., Cho, Y. K., and Suryanto, W. 2015. "Performance test of wireless technologies for personnel and equipment proximity sensing in work zones." Journal of Construction Engineering and Management. 142 (1): 04015049. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001031.

[77] Park, J.W., Marks, E., Cho, Y.K., and Suryanto, W. (2016). "Performance test of wireless technologies for personnel and equipment proximity sensing in work zones." Journal of Construction Engineering and Management, 142(1): 04015049.

[78] Park, J.W., Yang, X., Cho, Y.K., and Seo, J.W. (2017). "Improving dynamic proximity sensing and processing for smart work-zone safety." Automation in Construction, 84(2017), 111-120.

[79] Park, M.W. and Brilakis, I. (2016). "Continuous localization of construction workers via integration of detection and tracking." Automation in Construction, 72(2016), 129-142.

[80] Park, M.-W., and Brilakis, I. 2012. "Construction worker detection in video frames for initializing vision trackers." Automation in Construction. 28 (Dec): 15-25. https://doi.org/10.1016/j.autcon.2012.06.001.

[81] Park, M.W., Brilakis, I. (2012). "Construction worker detection in video frames for initializing vision trackers." Automation in Construction, 28, 15-25.

[82] Park, M.W., Makhmalbaf, A., and Brilakis, I. (2011). "Comparative study of vision tracking methods for tracking of construction site resources." Automation in Construction, 20(2011), 905-915.

[83] Patel, R. and Patel, S. 2020. "A comprehensive study of applying convolutional neural network for computer vision." International Journal of Advanced Science and Technology. 29(6s): 2161-2174.

[84] Pavlakos, G., Zhou, X., Derpanis, K.G., and Daniilidis, K. 2017. "Ordinal depth supervision for 3d human pose estimation." Conference on Computer Vision and Pattern Recognition (CVPR) 2018.

[85] Pavllo, D., Feichtenhofer, C., Grangier, D., and Auli, M. 2018. "3D human pose estimation in video with temporal convolutions and semi-supervised training." In Conference on Computer Vision and Pattern Recognition (CVPR) 2019.

[86] Pellegrini, S., Ess, A., and Van Gool, L. 2010. "Improving data association by joint modeling of pedestrian trajectories and groupings." In Proc., European conference on computer vision. 452-465. Crete, Greece: Springer.

[87] Pfeiffer, M., Paolo, G., Sommer, H., Nieto, J., Siegwart, R., and Cadena, C. 2018. "A data-driven model for interaction-aware pedestrian motion prediction in object cluttered environments." In Proc., 2014 IEEE International Conference on Robotics and Automation., 1-8. Brisbane, QLD, Australia: IEEE.

[88] Pratt, S.G., Fosbroke, D.E., and Marsh, S.M. (2001). "Building safer highway workzones: Measures to prevent injuries from vehicles and equipment." Department of Health and Human Services: Center for Disease Control and Prevention.

[89] Redmon J. and Farhadi, A. (2018). "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767.

[90] Redmon, J., and Farhadi, A. 2018. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767.

[91] Ren, S., He, K., Girshick, R. (2017). "Faster R-CNN: Towards real-time object detection with region proposal netowkrs." IEEE Transaction on Pattern Analysis and Machine Intelligence, 39, 1137-1149.

[92] Research and Markets. 2019. "Global construction robot market - drivers, restraints, opportunities, trends, and forecast up to 2025." (URL: https://www.researchandmarkets.com, accessed on Sept. 08 2019)

[93] Ruff, T. 2006. "Evaluation of a radar-based proximity warning system for off-highway dump trucks." Accident Analysis & Prevention. 38 (1): 92-98. https://doi.org/10.1016/j.aap.2005.07.006.

[94] Ruff, T.M. (2001). "Monitoring blind spots: A major concern for haul trucks." Engineering and Minining Journal, 202(12), 17-26.

[95] Salimans, T., and Kingma, D. P. 2016. "Weight normalization: A simple reparameterization to accelerate training of deep neural networks." In Proc., 30th Conference on Neural Information Processing Systems., 901-909. Barcelona, Spain: NIPS.

[96] SDLG Machine. (2014). “Reliability in Action: Backhoe Loader B877.” <http://www.sdlg-africa.com/wp-content/uploads/> (Aug. 22, 2017).

[97] Seo, J.O., Han, S.U., Lee, S.H., Kim, H.K. (2015). "Computer vision techniques for construction safety and health monitoring." Advanced Engineering Informatics, 29, 239-251.

[98] Sigal, L., Balan, A.O., and Black, M.J. 2010. "HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion." International Journal of Computer Vision (IJCV), 87(1-2):4.

[99] Tavares, P., Costa, C. M., Rocha, L., Malaca, P., Costa, P., Moreira, A. P., Sousa, A., and Veiga, G. .2019. "Collaborative Welding System using BIM for Robotic Reprogramming and Spatial Augmented Reality." Automation in Construction. 106 (Oct): 102825. https://doi.org/10.1016/j.autcon.2019.04.020.

[100] Tay, M. K. C., and Laugier, C. 2008. "Modelling smooth paths using gaussian processes." In Proc., Field and Service Robotics., 381-390.

[101] Teizer, J. (2015). "Wearable, wireless identification sensing platform: Self-monitoring alert and reporting technology for hazard avoidance and training (smarthat)." Electronic Journal of Information Technology in Construction, 20, 295-312.

[102] Teizer, J. 2015. "Wearable, wireless identification sensing platform: self-monitoring alert and reporting technology for hazard avoidance and training (SmartHat)." Journal of Information Technology in Construction. 20 (19): 295-312.

[103] Teizer, J., Allread, B. S., Fullerton, C. E., and Hinze, J. 2010. "Autonomous pro-active real-time construction worker and equipment operator proximity safety alert system." Automation in construction. 19 (5): 630-640. https://doi.org/10.1016/j.autcon.2010.02.009.

[104] Teizer, J., Allread, B.S., Fullerton, C.E., and Hinze, J. (2010). "Autonomous pro-active real-time construction worker and equipment operator proximity safety alert system." Automation in Construction, 19, 630-640.

[105] Teizer, J., and Vela, P.A. (2009). “Personnel tracking on construction sites using video cameras.” Advanced Engineering Informatics, 23(2009), 452-462.

[106] Tekin, B., Marquez Neila, P., Salzmann, M., and Fua, P. 2017. "Learning to fuse 2d and 3d image cues for monocular body pose estimation." In International Conference on Computer Vision (ICCV) 2017.

[107] Tractica. 2019. "Construction & demolition robots - robot assistants and structure, finishing, and infrastructure robots: global market analysis and forecast." (URL: https://www.tractica.com/research/construction-demolition-robots, accessed on Sept. 08 2019)

[108] Trautman, P., Ma, J., Murray, R. M., and Krause, A. 2015. "Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation." The International Journal of Robotics Research. 34 (3): 335-356. https://doi.org/10.1177/0278364914557874.

[109] Tsuruta, T., Miura, K., and Miyaguchi, M. 2019. "Mobile robot for marking free access floors at construction sites." Automation in Construction. 107 (Nov): 102912. https://doi.org/10.1016/j.autcon.2019.102912.

[110] United Nation (UN). 2021. <https://www.un.org/development/desa/en/news/population/world-population-prospects-2019.html> (Mar. 12, 2021)

[111] US Bureau of Labor Statistics (US BLS), United States Department of Labor. "Census of fatal occupational injuries (CFOI)." 2009-2018.

[112] Vähä, P., Heikkilä, T., Kilpeläinen, P., Järviluoma, M., and Gambao, E. 2013. "Extending automation of building construction—Survey on potential sensor technologies and robotic applications." Automation in Construction. 36 (Dec): 168-178. https://doi.org/10.1016/j.autcon.2013.08.002.

[113] Varghese, J. Z., and Boone, R. G. 2015. "Overview of autonomous vehicle sensors and systems." In Proc., International Conference on Operations Excellence and Service Engineering., 178-191.

[114] Vega-Heredia, M., Mohan, R. E., Wen, T. Y., Siti'Aisyah, J., Vengadesh, A., Ghanta, S., and Vinu, S. 2019. "Design and modelling of a modular window cleaning robot." Automation in Construction. 103 (Jul): 268-278. https://doi.org/10.1016/j.autcon.2019.01.025.

[115] Verified Market Research (VMR). 2018. <https://www.verifiedmarketresearch.com/product/construction-robot-market/> (Mar. 12,

2021)

[116] Waehrer, G.M., Dong, X.S., Miller, T., Haile, E., and Men, Y. 2007. "Costs of occupational injuries in construction in the united states." Accident Analysis & Prevention, 39(6): 1258-1266.

[117] Wang, M.-z., Luo, M., Cen, Y.-w., and Huang, J.-z. 2018. "Research on Space Pose and Hydraulic System Stability of Remote-Controlled Demolition Robot." In Proc., 5th International Conference on Information Science and Control Engineering., 962-967.

[118] Wefelscheid, C., Hansch, R., and Hellwich, O. (2011). "Three-dimensional building reconstruction using images obtained by unmanned aerial vehicles." International Conference on Unmanned Aerial Vehicle in Geomatics, Zurich, Switzerland.

[119] Więckowski, A. 2017. "“JA-WA”-A wall construction system using unilateral material application with a mobile robot." Automation in Construction. 83 (Nov): 19-28. https://doi.org/10.1016/j.autcon.2017.02.005.

[120] Wu, J., Cai, N., Chen, W., Wang, H., and Wang, G. 2019. "Automatic detection of hardhats worn by construction personnel: A deep learning approach and benchmark dataset." Automation in Construction. 106(2019): 102894.

[121] Xu, Y., Piao, Z., and Gao, S. 2018. "Encoding crowd interaction with deep neural network for pedestrian trajectory prediction." In Proc., IEEE Conference on Computer Vision and Pattern Recognition., 5275-5284. Salt Lake City, UT: IEEE.

[122] Yamaguchi, K., Berg, A. C., Ortiz, L. E., and Berg, T. L. 2011. "Who are you with and where are you going?" In Proc., IEEE Conference on Computer Vision and Pattern Recognition., 1345-1352. Colorado Springs, CO: IEEE.

[123] Yang, J., Arif, O., Vela, P.A., Teizer, J., and Shi, Z. (2010). "Tracking multiple workers on construction sites using video cameras." Advanced Engineering Informatics, 24, 428-434.

[124] Yang, M.D., Chao, C.F., Huang, K.S., Lu, L.Y., and Chen, Y.P. (2013). "Image-based 3D Scene Reconstruction and Exploration in Augmented Reality." Automation in Construction, 33(2013) 48-60.

[125] Yang, Y., Pan, M., and Pan, W. 2019. "Co-evolution through interaction’of innovative building technologies: The case of modular integrated construction and robotics." Automation in Construction. 107 (Nov): 102932. https://doi.org/10.1016/j.autcon.2019.102932.

[126] Ye, S., Nourzad, S., Pradhan, A., Bartoli, I., and Kontsos, A. (2014). “Automated detection of damaged areas after hurricane sandy using aerial color images.” Computing in Civil and Building Engineering (2014), Reston, VA. USA.

[127] Yu, S. N., Lee, S. Y., Han, C. S., Lee, K. Y., and Lee, S. H. 2007. "Development of the curtain wall installation robot: Performance and efficiency tests at a construction site." Autonomous Robots. 22 (3): 281-291.

[128] Zhang, C. and Elaksher, A. (2012). "An unmanned aerial vehicle-based imaging system for 3D measurement of unpaved road surface distresses." Computer-Aided Civil and Infrastructure Engineering, 27(2), 118–129.

[129] Zhu, Z., Ren, X., and Chen, Zhi. (2017). "Integrated detection and tracking of workforce and equipment from construction jobsite videos." Automation in Construction, 81(2017), 161-171/

[130] Zollmann, S., Hoppe, C., Kluckner, S., Poglitsch, C., Bischof, H., and Reitmayr, G. (2014). "Augmented reality for construction site monitoring and documentation." Poceedings of the IEEE, 102(2), 137–154.

[131] 3D Pose in the Wild (3DPW) Challenge. <https://virtualhumans.mpi-inf.mpg.de/3DPW_Challenge/> (Mar. 13, 2021).